Advances in Computational Linguistics

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 90,** mayo 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 90**, May 2015. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 90

Advances in Computational Linguistics

Alexander Gelbukh,
Samhaa R. El-Beltagy (eds.)

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
“La Técnica al Servicio de la Patria”

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

We are honored to present to the reader a special issue of *Research in Computing Science* dedicated to a rapidly developing area of computational linguistics, at the crossroads of computer science, artificial intelligence, and linguistics.

Computational linguistics, also known as natural language processing, plays an important role in the development of modern computer science and has numerous industrial applications. Such industrial giants as Microsoft, Google, and Apple view it as a crucial area of development of user interfaces and numerous information services.

Traditional application areas of natural language processing include machine translation and information retrieval. Nowadays, these areas come in many flavor and with far-reaching extensions. The classical machine translation has evolved into multilingual and cross-lingual natural language processing. Modern state of computational linguistics is characterized by serious attention to world's languages, both minor languages as well as major but under-resourced languages. Accordingly, this volume includes a selection of papers devoted to various aspects of Arabic, Amazigh, Chinese, Romanian, Turkish, Yoruba, and various Indian languages, among others.

On the other hand, information retrieval has evolved into many application areas such as question answering or text summarization. These areas are also represented in this special issue.

In addition, new applications of natural language processing constantly arise. One of such new application fields is sentiment analysis and opinion mining, which leverages the power of social networking and user-generated content for large-scale analysis of opinions the people express in Internet. Such analysis improves the income of companies by suggesting insights into users' acceptance of their products, improves democracy by providing the governments and political parties with real-time feedback from the citizens, and improves ordinary people's quality of life by fueling collaborative recommender systems and providing the consumers with statistics of experience of other consumers with products and services of interest. This volume includes a number of papers on sentiment analysis and opinion mining.

The papers included in this special issue were selected basing on a double-blind review procedure, with participation of 132 leading experts in the topic from 37 countries, listed at the end of this volume.

This special issue will be useful to researchers, students, and engineers working in natural language processing, human language technologies, and computational linguistics.

Alexander Gelbukh,
Research Professor, Instituto Politécnico Nacional, Mexico

Samhaa R. El-Beltagy,
Professor, Nile University, Egypt

Guest Editors

May 2015

Table of Contents

Page

Expectation Maximization Algorithm for Domain Specific Ontology Extraction

Brijesh Bhatt and Pushpak Bhattacharyya

Center for Indian Language Technology
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
brij.s.bhatt@gmail.com pb@cse.iitb.ac.in

Abstract. Learning ontology from unstructured text is a challenging task. Over the years, a lot of research has been done to predict ontological relation between a pair of concepts. However all these measures predict relation with a varying degree of accuracy. There has also been work on learning ontology by combining evidences from heterogeneous sources, but most of these algorithms are ad hoc in nature. In this paper we investigate wide range of evidences to predict relation between a pair of concepts and propose a standardized Expectation Maximization algorithm to construct domain specific ontology. The proposed approach is completely unsupervised and does not require any seed terms or human intervention. In addition, the proposed approach can also be easily adopted for any language. We have conducted our experiments for two languages Hindi and English and for two domains Health and Tourism. The average F-Score observed in all experiments is above 0.60.

1 Introduction

Ontology is defined as 'Explicit specification of conceptualization' [13]. As a knowledge representation formalism, ontologies have found a wide range of applications in the areas like knowledge management, information retrieval and information extraction. Domain and application specific ontologies play a crucial role in semantic web applications.

As manual construction of ontology is a cumbersome task, a lot of research is being done to automatically construct ontology from the unstructured text. In general, ontology learning process involves two basic tasks- domain specific concept identification and construction of concept hierarchy. Most of the existing algorithms extract relevant terms from the documents using various term extraction methods [19, 23, 10, 11] and then construct ontology by identifying subsumption relations between terms.

Detection of subsumption relation is a core task of ontology extraction. Over the years, a number of approaches have been proposed to detect subsumption between a pair of concepts. These approaches can be divided into three categories: Pattern based, Knowledge based, and statistical. Pattern based approaches rely

pp. 11–21; rec. 2015-01-25; acc. 2015-02-25

on seminal work of [15] who outlined a variety of lexico-syntactic patterns that can be used to find out 'IS-A' from a text. Statistical approaches are based on distributional hypothesis that is 'similar terms appear in the similar context' [14]. Availability of general purpose lexical resources like, WordNet [9], Wikipedia etc. gave rise to knowledge based approach. Many researchers have used *Hypernymy* relation of WordNet and category pages of Wikipedia to detect subsumption relation between a pair of concepts [8].

While all these approaches detect subsumption relation with a reasonable precision, they are quite different and have their own strengths and weaknesses. Pattern based approach relies on language specific patterns and hence does not cater to different languages. Patterns are also not very frequent so this approach may not work well for a small corpus. Statistical approaches primarily detects 'semantic relatedness' between a pair of concept but detection of relation still remains a challenging task. Besides, the result of these approaches are often contradictory.

There has been efforts to learn domain ontology by combining different approaches but these methods are mostly ad hoc. In addition, most of the existing work in ontology learning is done for English language and often uses sophisticated Natural Language Processing (NLP) tools and resources. In absence of such tools and resources, it becomes difficult to adopt these methods for resources constrained languages.

The key challenges in the task of ontology learning are as follows,

- Development of a standardized approach to combine different subsumption detection technique for better ontology extraction
- Development of an approach which can be easily adopted for different languages; particularly resource constrained languages

In order to address the above mentioned challenges, we propose an ontology extraction framework which learns ontology from heterogeneous sources and does not rely on sophisticated NLP tools and resources. In this paper, we first analyze various techniques for subsumption detection and then propose an Expectation-Maximization algorithm to learn ontology. The key contributions of our work are as follows,

- The proposed system is completely unsupervised and does not require any labeled data or human intervention
- The proposed approach does not rely on any language specific technique or resource and hence can be adopted for any language

The remaining of the paper is organized as follows, section 2 presents a survey of existing work, section 3 describe the algorithm to learn ontology, section 4 describes experiments and observations and section 5 provides.

2 Related Work

Ontology learning approaches can be divided into three categories: heuristic based, statistical and hybrid techniques. Heuristic approach [15, 2, 12] primarily relies on the fact that ontological relations are typically expressed in language via a set of linguistic patterns. [15] outlined a variety of lexico-syntactic patterns that can be used to find out ontological relations from a text. She described a syntagmatic technique for identifying hyponymy relations in free text by using frequently occurring patterns like '*NP0 such as NP1, NP2, ,NPn*'. [2] used a pattern-based approach to find out part-whole relationships (such as between car and door, or car and engine) in a text. Heuristic approaches rely on language-specific rules which cannot be transferred from one language to another.

Statistical approaches model ontology learning as a classification or clustering problem. Statistical methods relate concepts based on distributional hypothesis [14], that is 'similar terms appear in the similar context.' [17] performed semantic clustering to find semantically similar nouns. They calculated the co-occurrence weight for each verb-subject and verb-object pair. Verb-wise similarity of two nouns is calculated as the minimum shared weight and the similarity of two nouns is the sum of all verb-wise similarities. [21] proposed a divisive clustering method to induce noun hierarchy from an encyclopedia.

Hybrid approaches leverage the strengths of both statistical and heuristic based approaches and often use evidences from existing knowledge bases such as WordNet, Wikipedia, etc. [3] combined the lexico-syntactic patterns and distributional similarity based methods to construct ontology. Similarity between two nouns is calculated by computing the cosine between their respective vectors and used for hierarchical bottom-up clustering. Hearst-patterns are used to detect hypernymy relation between similar nouns. In a similar approach, [5] clustered nouns based on distributional similarity and used Hearst-patterns, WordNet [9] and patterns on the web as a hypernymy oracle for constructing a hierarchy. Unlike [3], the hypernymy sources are directly integrated into the clustering, deciding for each pair of nouns how they should be arranged into the hierarchy. [8] used Wikipedia to extract ontology for different languages.

Most of the hybrid approaches which combine pattern based approach and statistical approach, are ad-hoc in nature. They first use the statistical clustering to group the terms and then uses knowledge base (e.g. WordNet) and patterns to identify relation. In this work, we are proposing a starndardized Expectation Maximization algorithm that merges evidences from different sources. We treat each measure as a feature to detect relation ship. Apart from these, the proposed algorithm detects three relations, synonymy, hypernymy (subsumption) and neighbor (co-hyponymy).

3 Algorithm

Most of the existing algorithms to learn ontology from heterogeneous sources of evidences are ad hoc in nature and use languages specific measures (e.g. lexical patterns for English, English WordNet etc) to detect relation between a pair of concepts. We also follow hybrid approach where we combine statistical, pattern based and knowledge based methods to learn ontology; but unlike other existing systems we choose evidences which can be adopted for any languages and we use a standardized Expectation-Maximization based algorithm which can be adopted for different sets of evidences. More specifically, we use evidences from different sources as features and use Expectation-Maximization algorithm to learn relation between a pair of concepts. The process of ontology learning is as described in Fig 1

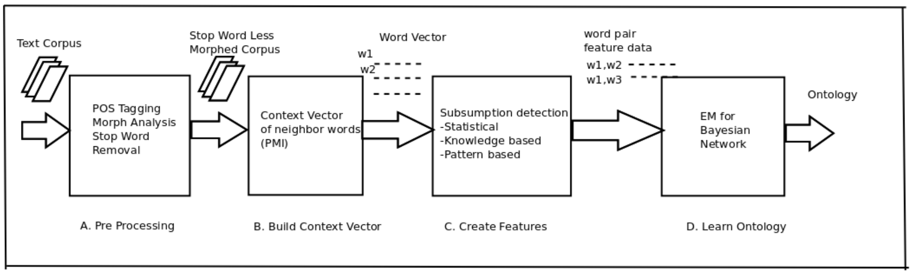


Fig. 1: Ontology Learning Process

3.1 Pre-Processing

The input text is processed using POS tagger and morph analyzer. The stop words and junk words are removed. The output of pre-processing step is pos tagged, stop word less corpus. For English we have used *morpha* [20] morph analyzer and Stanford POS tagger [1], for Hindi we have used morph analyzer and pos tagger developed at IITB [2].

3.2 Context Vector Construction

Key terms from the corpus are extracted using pattern based method. Lexical pattern $(NP)^*(NP)$ is applied to extract key phrases from the corpus. Relevance of the key term is calculated by counting the frequency of the term. Terms are filtered out using weirdness measure [1].

In order to compare concepts, we construct the context for a word using bag of word approach. Feature vector for each term is created by including co-occurring nouns, verbs and adjectives. Co-occurrence is calculated using Point-wise Mutual Information [4] measure.

3.3 Feature Construction

Features are the indicator of semantic relation between a pair of words. In order to construct concept hierarchy we detect subsumption (hypernymy), synonymy and co-hyponymy relations. Various measures we used to detect these relations are as shown in Table 1.

[1] http://nlp.stanford.edu/software/tagger.shtml
[2] http://www.cfilt.iitb.ac.in/Tools.html

Table 1: Features for ontology Learning

	feature	Description	formula
f_1	Cosine Similarity	Cosine similarity between word w_1 and w_2 is calculated by comparing the vectors of words.	$cosine(w_1, w_2) = \frac{\overrightarrow{V}(w_1) \cdot \overrightarrow{V}(w_2)}{\lvert\overrightarrow{V}(w_1)\rvert\lvert\overrightarrow{V}(w_2)\rvert}$
f_2	Weeds Precision	This measure quantifies the weighted inclusion of the features of a term w_1 within the features of a term w_2. [24, 18].	$WeedsPrec(w_1, w_2) = \frac{\sum_{f \in F(w_1) \bigcap F(w_2)} w_1(f)}{\sum_{f \in F(w_1)} w_1(f)}$
f_3	cosWeeds	This measure corresponds to the geometrical average of Weeds Precision and cosine similarity between words w_1 and w_2	$cosWeeds = \sqrt{cosine(w_1, w_2). WeedsPrec(w_1, w_2)}$
f_4	ClarkeDE	This measure is a close variation of Weeds Precision, proposed by [6].	$ClarkeDE(w_1, w_2) = \frac{\sum_{f \in F(w_1) \bigcap F(w_2)} min(w_1(f), w_2(f))}{\sum_{f \in F(w_1)} w_1(f)}$
f_5	Frequency Ratio	We use frequency ratio to measure degree of generality of a word. The measure is based on following hypothesis, 'A more general term appears more frequently in the corpus, while a more specific term appears less frequently' [22]	$fratio(w_1, w_2) = \frac{f(w_1)}{f(w_2)}$
f_6	Head Word heuristic Pattern	This pattern finds hypernymy relation from noun phrase. e.g. "Heritage Hotel" is a "Hotel"	(NP)*NP is hyponym of (NP)
f_7	Neighbor Pattern	This pattern detects neighbor (Co-hyponymy) relation. e.g. Delhi, Mumbai, Calcutta are cities.	$((NP)*(NP)(CC\|,))*(NP)$
f_8	WordNet hypernym	This formula calculates probability of hypernymy by consulting WordNet	$\frac{hypernym(w_1, w_2)}{totalRelation(w_1, w2)}$
f_9	WordNet Synonym	This formula calculates probability of synonymy by consulting WordNet	$\frac{synonym(w_1, w_2)}{totalRelation(w_1, w2)}$
f_{10}	WordNet Neighbor	This formula calculates probability of co-hyponymy by consulting WordNet	$\frac{co-hyponym(w_1, w_2)}{totalRelation(w_1, w2)}$

3.4 Ontology Learning

Output of feature construction step is as shown in Table 2. As shown in Table 2, the output of different features does not match and they often contradict. Our aim is to predict the correct relation between the pair of words using the observed values of features. Each of this feature gives a hint about possible relation between a pair of words. We assume that the relation y is the common cause that triggers one or more features. We model a Bayesian network as shown in the figure 2. The random variable Y corresponds to the relation between the pair of words and $X_1...X_d$ correspond to the feature vector components $f_1, f_2, ..., f_d$. $X_1...X_d$ are observed variables and Y is the hidden variable e.g. the relation that we want to predict.

Table 2: Example DataSet

i	**Word Pair**	$\mathbf{f_1}$	$\mathbf{f_2}$	$\mathbf{f_3}$	$\mathbf{f_4}$	$\mathbf{f_5}$	$\mathbf{f_6}$	$\mathbf{f_7}$	$\mathbf{f_8}$	$\mathbf{f_9}$	$\mathbf{f_{10}}$	$\mathbf{f_{11}}$	$\mathbf{f_{12}}$	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_3}$
1	haemorrhagic fever - dengue haemorrhagic fever	1	1	1	1	1	1	0	0	0	0	0	0	?	?	?
2	leptospirosis-kalaazar	2	0.39	0.24	0.24	0.30	0	0	1	0	0	0	0	?	?	?
3	transplant - transplantation	1.91	0.20	0.14	0.14	0.17	0	0	0	0.33	0	0	0	?	?	?
.	...	...	...	...	...	...	...	...	...	...	...	...	....	?	?	?
n	cannabis - marijuana	1.6	0.18	0.14	0.14	0.16	0	0	0	0.25	0	0.25	0	?	?	?

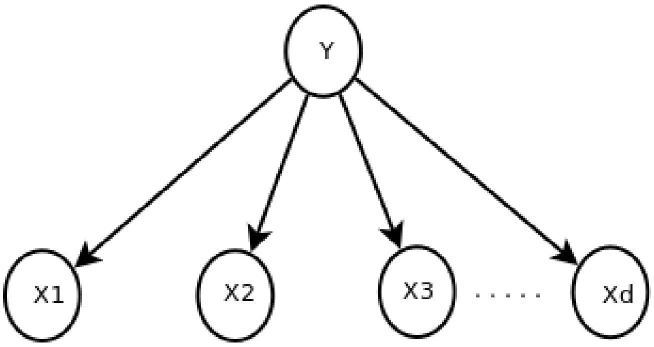


Fig. 2: Bayesian Model for Predicting Relation

The problem of predicting correct relation between a pair of word can now be posed as a Bayesian network learning problem. Given the observed variables X (features) our aim is to predict the hidden variable Y (relation e.g. synonymy, hypernymy, co-hyponymy).

Joint probability of the network can be expressed as shown in equation 1.

$$P(Y=y,X_1=x_1,X_2=x_2,...X_d=x_d)=P(Y=y)\prod_{j=1}^{d}P(X_j=x_j|Y=y) \tag{1}$$

Here, $P(Y=y)$ and $P(X_j=x_j|Y=y)$ for $j=1,2...,d$ are network parameters. Let's define parameter vector θ as a vector consisting values of these parameters.

Likelihood Since the value of hidden variable Y (relation between word pair) is not known, probability of an example can be calculated by marginalizing over all possible values of label Y, as shown in the following equation 2.

$$P(x)=\sum_{y=1}^{k}P(Y=y)\prod_{j=1}^{d}P(X_j=x|Y=y) \tag{2}$$

For the complete training set containing n examples, likelihood can be calculated using the equation 3.

$$L(\theta)=\sum_{i=1}^{n}log\sum_{y=1}^{k}(P(Y^{(i)}=y)\prod_{j=1}^{d}P(X_j^{(i)}=x|Y^{(i)}=y)) \tag{3}$$

Our goal is to predict the network parameters θ that maximize likelihood of the data.

EM Algorithm Since the relation label is not known, we use Expectation-Maximization algorithm to iteratively estimate the value of parameters θ that maximizes the likelihood. For the training set consisting of vectors of observed variable $x^{(i)}$ and hidden variables $y^{(i)}$ for $i=1...n$ and the parameter vector θ, the EM algorithm starts by randomly choosing the initial parameter values θ_0. At each iteration value of hidden variable $Y^{(i)}$ is calculated as a function of the training set and the previous parameter values θ_{t-1}; and the new parameter values θ_t are updated using the observed variables and previously estimated hidden variables. [16, 7].

E-Step For the given value of θ, E-Step calculates probability of hidden variable for each example X_i using equation 5.

$$\delta(y|i) = p(Y^{(i)} = y|X^{(i)}; \theta_{t-1}) \tag{4}$$

$$p(Y^{(i)} = y|x^{(i)}; \theta_{t-1}) = \frac{P(Y^{(i)} = y) \prod_{j=1}^{d} P(X_j^{(i)} = x_j/Y^{(i)} = y)}{\sum_{y=1}^{k} \left(P(Y^{(i)} = y) \prod_{j=1}^{d} P(X_j^{(i)} = x_j/Y^{(i)} = y) \right)} \tag{5}$$

Expectation of hidden variable Y is then calculated using equation 6

$$E(Y = y) = \sum_{i=1}^{n} \delta(y|i) \tag{6}$$

M-Step Based on the value of hidden variable estimated in E-Step, M-Step calculates new parameter values as shown in equation 7 and 9.

$$P(Y = y)^t = \frac{\sum_{i=1}^{n} \delta(y|i)}{n} \tag{7}$$

$$where, n = totalnumberofexamples \tag{8}$$

$$p(X_j = x|Y = y)^t = \frac{\sum_{i=1:X_j^i=x}^{n} \delta(y|i)}{\sum_{i=1}^{n} \delta(y|i)} \tag{9}$$

4 Experiments and Observations

We have carried out our experiments for two domains, health and tourism and for two languages, English and Hindi. We choose English to compare results against benchmark and We choose Hindi as a resource constrained language. We have used untagged text corpus for both domains Health and Tourism [3]. Table 3 shows the details of the corpora.

We preprocessed the English corpus using Stanford POS tagger and Morpha morph analyzer and Hindi corpus with CFILT POS Tagger. After extracting key terms and building context vector the features are extracted as described earlier.

In order to measure performance of Individual features we measure precision of top 100 word pair for each measures. Table 4 summarizes the results. As shown in the table pattern based and Knowledge based measures perform much better than the statistical measures. However, the relation tuple detected by all these measures are often different. WordNet based measures detect subsumption between more frequent and general terms while pattern based and statistical measures detect relation between low frequency terms.

[3] The corpora are available at $http://www.cfilt.iitb.ac.in/wsd/annotated\_corpus/$

Table 3: corpus details

Domain	Language	Total Sentences	Words
Health	Hindi	25000	61000
	English	25000	69000
Tourism	Hindi	48000	89000
	English	59000	121000

Table 4: precision@100 for Individual features

Corpus	Weeds Precision	cosWeeds	clarkeDE	Pattern	WordNet
English Health	0.36	0.39	0.32	.67	.72
English Tourism	0.42	0.48	0.48	.69	.70
Hindi Health	0.34	0.40	0.38	.70	.68
Hindi Tourism	0.40	0.38	0.36	0.65	0.68

The quality of the ontology constructed is evaluated by comparing it with the hand crafted ontology. The lexical precision and recall is calculated using following formula,

Recall = $|y_1^r \bigcap y_2^r|/y_2^r$ and Precision = $|y_1^r \bigcap y_2^r|/y_1^r$

where y_1^r is the set of relation r detected in automatically constructed ontology and y_2^r is the set of relation r detected in hand crafted gold standard. We have run experiments to detect four relations i.e. *hypernymy, synonymy, neighbors (co-hyponymy) and no relation.* The precision (P), recall (R) and F-Score (F) for each domain and for each language are summarized in the table 5

Table 5: Experiment Results

Domain	Language	Hypernymy			Neighbor			Synonym			No Relation			Average		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Health	Hindi	0.59	0.84	0.69	0.66	0.65	0.66	0.42	0.81	0.55	0.82	0.46	0.59	0.62	0.69	0.66
	English	0.62	0.85	0.72	0.61	0.70	0.65	0.58	0.75	0.66	0.81	0.50	0.62	0.66	0.70	0.68
Tourism	Hindi	0.54	0.85	0.66	0.62	0.60	0.61	0.3	0.63	0.41	0.8	0.43	0.55	0.57	0.63	0.6
	English	0.56	0.79	0.65	0.54	0.65	0.59	0.63	0.75	0.68	0.76	0.47	0.58	0.62	0.66	0.64

As shown in table 5, the average F-Score observed for both languages and both domains is between 0.60 to 0.70. The performance is reasonably good considering the fact that the algorithm is completely unsupervised and does not rely on language specific evidences. Synonymy detection does not perform very well for Hindi language as resources for Hindi are not as rich as that of English.

5 Conclusion

In this paper we have presented an unsupervised algorithm for domain ontology extraction. The proposed approach does not rely on language specific resources or tools and can be easily adopted for any language. The proposed algorithm consults evidences from different sources e.g. statistical measures, knowledge based measures and pattern based measures and predicts relation between a pair of words. While most of the existing ontology learning algorithms focuses only on hypernymy / IS-A relation detection, our algorithm also detects synonymy and co-hyponymy, thus it provides a more refined ontology by merging words that represent similar concepts. The proposed Expectation-Maximization framework is general enough to accommodate other measures as features or to detect more relations (e.g. whole-part etc). We have conducted experiments for two languages and two domains and average precision and recall was higher than 0.60.

References

1. Ahmad, K., Gillam, L., Tostevin, L., Group, A.: Weirdness indexing for logical document extrapolation and retrieval (wilder). In: The Eighth Text REtrieval Conference (1999)
2. Berland, M., Charniak, E.: Finding parts in very large corpora. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 57–64. ACL '99, Association for Computational Linguistics, Stroudsburg, PA, USA (1999)
3. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics. pp. 120–126 (1999)
4. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. 16(1), 22–29 (mar 1990)
5. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning taxonomic relations from heterogeneous sources of evidence. Ontology Learning from Text: Methods, Evaluation and Applications (2005)
6. Clarke, D.: Context-theoretic semantics for natural language: An overview. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics. pp. 112–119. GEMS '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
7. Collins, M.: The naive bayes model, maximum-likelihood estimation, and the em algorithm (2013), lecture notes
8. Domínguez García, R., Schmidt, S., Rensing, C., Steinmetz, R.: Automatic taxonomy extraction in different languages using wikipedia and minimal language-specific information. In: Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing - Part I. pp. 42–53. CICLing'12, Springer-Verlag, Berlin, Heidelberg (2012)
9. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)

10. Frantzi, K.T., Ananiadou, S., Tsujii, J.i.: The c-value/nc-value method of automatic recognition for multi-word terms. In: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries. pp. 585–604. ECDL '98, Springer-Verlag, London, UK, UK (1998)
11. Gacitua, R., Sawyer, P., Gervasi, V.: Relevance-based abstraction identification: technique and evaluation. Requir. Eng. 16(3), 251–265 (sep 2011)
12. Girju, R., Badulescu, A., Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of HLT/NAACL-03. pp. 80–87 (2003)
13. Gruber, T.R.: Towards principles for the design of ontologies used for knowledge sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Deventer, The Netherlands (1993)
14. Harris, Z.: Mathematical structures of language. John Wiley Sons (1968)
15. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics. pp. 539–545 (1992)
16. Heckerman, D.: Learning in graphical models. chap. A Tutorial on Learning with Bayesian Networks, pp. 301–354. MIT Press, Cambridge, MA, USA (1999)
17. Hindle, D.: Noun classification from predicate-argument structures. In: Proceedings of the 28th annual meeting on Association for Computational Linguistics. pp. 268–275. ACL '90, Association for Computational Linguistics, Stroudsburg, PA, USA (1990)
18. Kotlerman, L., Dagan, I., Szpektor, I., Zhitomirsky-geffet, M.: Directional distributional similarity for lexical inference. Nat. Lang. Eng. 16(4), 359–389 (oct 2010)
19. Kozakov, L., Park, Y., Fin, T.H., Drissi, Y., Doganata, Y.N., Cofino, T.: Glossary extraction and utilization in the information search and delivery system for ibm technical support. IBM Systems Journal 43(3), 546–563 (2004)
20. Minnen, G., Carroll, J., Pearce, D.: Applied morphological processing of english. Nat. Lang. Eng. 7(3), 207–223 (sep 2001)
21. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of english words. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics. pp. 183–190 (1993)
22. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intell. Res. (JAIR) 11, 95–130 (1999)
23. Sclano, F., Velardi, P.: Termextractor: a web application to learn the shared terminology of emergent web communities (2007)
24. Weeds, J., Weir, D.: A general framework for distributional similarity. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 81–88. EMNLP '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003)

Projecting LMF Lexica Towards OWL-DL through LMF-JAPE Patterns to Obtain Interoperable Formats

Lhioui Malek[1], Kais Haddar[2], Laurent Romary[3]

[1] Laboratoire MIRACL, Multimedia, InfoRmation Systems and Advanced Computing Laboratory, Université de Sfax, Tunisie
[2] Laboratoire MIRACL, Multimedia, InfoRmation Systems and Advanced Computing Laboratory, Université de Sfax, Tunisie
[3] Inria & Centre Marc Bloch, Berlin, Germany

ma.lhioui@gmail.com, kais.haddar@yahoo.fr,
laurent.romary@inria.fr

Abstract. The development of editors, analyzers, translators and other NLP system types can involve several representation languages. The heterogeneity of representation languages induces the interoperability issue at different levels and in different contexts. In language technology, interoperability proved very crucial nowadays since its lack costs the translation industry a fortune where it is paid primarily for adjusting data formats. With this regard, we consider that representing LMF (Lexical Markup Framework: ISO-24613) lexica in OWL-DL (Web Ontology Language Description Logic) can be a serious attempt to achieve these goals. In this paper, we study the requirements of this proposal. We formulate an OWL ontology variant by explaining LMF mapping process to OWL version. The evaluation of the OWL variant construction of the LMF process is measured using the instantiation of the OWL-DL ontology.

Keywords: Interoperability, LMF, OWL-DL, LMF-JAPE patterns.

1 Introduction

The reusability notion in the past has evolved nowadays into interoperability. “This notion means the ability of information and communication systems to exchange data and to enable the sharing of information and knowledge” (Francopoulo, 2013).

Projects today require such a strategy, and to play the role of the keystone in many domains must rely on interoperability, otherwise they are out of business. A new article by TAUS (TAU, 2011) declares that: “The lack of interoperability costs the translation industry a fortune”. In fact, this fortune is compensated mostly in order to adjust data formats. Interoperability concept can solve sharing problems based on involved elements semantics. For example, in the web field, the links between pages have no direct bearing managed by machines. "The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee et al., 2001).

Unlike other areas, the NLP has not recognized an expansion of interoperability concept. The first serious attempts to this notion were after the development of LMF. In fact, standards facilitate greatly the construction of a powerful interoperability strategy. In fact, it provides well-structured lexical resources that are able to be open and shared after relevant mapping. Ontologies are one of the very recognized mapping activities. Indeed, these activities have proven their interest through many academic researches: WordNet or FrameNet are developed to disambiguate the semantic side of the terms or incorporate them into other ontology (Assem et al., 2006).

In this paper, we propose a rigorous method strictly founded to manage the projection from LMF to OWL format as a serious attempt to succeed interoperability between existing data formats particularly LTAG, HPSG, TEI and LMF[1]. This method can be used further in order to operate the transitivity between different existing data formats. In fact, there are several works proposing the mapping LTAG-HPSG[2], HPSG-LMF. So, by this projection LMF-OWL and using the transitivity notion between all the mentioned formats, the mapping between LTAG-OWL and HPSG-OWL will be operational.

We present in this paper an ontological OWL variant detailing the process of transforming the XML to OWL and we can prove its usefulness in NLP field by illustrating a reflection of morphosyntactic annotation.

2 Related Works and General Context

The ontologies construction requires the presence of well-structured methods. Generally, these construction methods are divided on statistical and linguistic approaches (Buitelaar et al., 2005). However, the two approach kinds require two extraction types: concepts and relations.

Statistically, concepts extraction requires analysis of co-occurring terms by studying their distribution or with probability determination. In a second step, relations extraction is the second key step in the construction process that can be determined using similarities between concepts (Grefenstette, 1994). Another method for predicting relations can also be identified using Bayesian networks (Weissenbacher and Nazarenko, 2007) or Text Mining techniques (Grcar et al., 2007).

In order to enrich ontologies, linguistic approaches were designed to collect association rules which are able to identify concepts and relations. Parsers are used in this case (Bourigault, 2002). Other methods may also be used to extract linguistic fingerprints of semantic relations to determine lexical and syntactic patterns. Other recent studies are leaded to enrich classic methods in order to separate content from structure. Therefore, recent attempts lead first to build the ontology core, then to develop them taking advantage from external resources through a preliminary analysis.

[1] LTAG : Lexicalized Tree Adjoint Grammar, HPSG : Head-driven Phrase Structure Grammar, TEI: Text Encoding Initiative

[2] From LTAG to HPSG

In the other side, nowadays, many works consisting on mapping from one format to another have been done. We can mention the mapping process already done by (Wilcock, 2007). This work presents an OWL ontology for HPSG. Another mapping process consists on projecting HPSG syntactic lexica towards LMF (Haddar et al., 2012). In the same context, a rule-based system has been created in order to translate LMF syntactic lexicon into TDL within the LKB[3] platform (Loukil et al., 2010). Furthermore, an owl-DL ontology has been conceived from an LMF input lexicon (Lhioui et al., 2014).

Until now, there are very few standards dedicated to the construction of normalized lexicons if we compare them with standards available for the manipulation of linguistic resources in general. However, the necessity of the construction of normalized lexicons is until now a hard task to be achieved. In fact, the normalization requires particular time resources with no left of human ones which are able to certify compatibility with chosen standards. LMF, which is conceived as a NLP standard, aims at covering large range of many languages. Consequently, having conformity to this standard makes our work in comparison with similar works on a global scale.

Because of its importance, we propose an initiative able to transform lexicon compliant LMF into an ontological OWL-DL variant. This allows supporting the development of reusable lexical data bases and then searching in the field of interoperability in future works.

Since the normalization identify necessary an information common coverage for all lexicons, the built coverage will be fundamental for many tools aiming the exchange and sharing of lexical resources and therefore provide the basis for developing an interoperable framework for with this type of ontological variant, the concept of interoperability: with this kind of ontological variant, interoperability notion will be able to be applied to exchange data and to enable knowledge sharing (Francopoulo, 2013). It will be a mixture of standards and guidelines such as the TEI (Sperberg-McQeen and Burnard, 2014). Thus, the standards will be systematically correlated and guidelines will explain the specification of these standards. Nowadays, having transformation prototype from LMF to OWL is very advisable. This prototype must have a big number of features, which will be explained thereafter.

3 Formalization of the Transformation Process

The transformation process requires a formalization step before its development. In order, to build a rigorous method allowing the transformation from LMF to OWL, a set of steps have to be formalized before any development process.

We start by giving an overview concerning LMF. Then, we present the conceptualization phase. Finally, we motivate this section by representing the construction phase describing the suitable formalisms used in this step.

[3] TDL: Type Description Language, LKB: Language Knowledge Builder

3.1 LMF Overview

After serious activities developed on building lexicons in teamworks, a group of 60 researchers was behind LMF standard creation (Francopoulo, 2013). LMF is an ISO standard which includes monolingual and multilingual lexica. LMF specification follows UML modeling Object Management Group (OMG). LMF is arranged on two big parts: core model and extensions packages. The LMF modelling principles take up the ISO committee TC 37 principles and let a lexical database designer to mix any component of the LMF meta-model with data-categories (Ide and Romary, 2004) in order to create an appropriate model. Data categories behave as UML attribute-value pairs in the diagrams. The core model contains the backbone of a lexical entry. It identifies critical concepts of vocabulary, word, form and sense. LMF core model is characterized by a hierarchical structure involving on several components.

3.2 Conceptualization Phase

In order to move from the source schema for a diagram source (whatever its nature: text, XML, etc.) to the ontology, the majority of construction methods design a conceptualization step in order to ensure the passage from the first plan to the second. The source in this article includes XML files. Conceptualization requires an analysis of the source tags. This preliminary analysis will show its interest in building the bases of ontology concepts and relations. For this reason, all classes of LMF package must be provided in advance an XML form.

Conceptualization requires prior analysis of properties and relations between LMF classes. This explains the creation of an XML file containing properties in the LMF classes that must be offered first.

3.3 Construction Phase

In order to develop the construction phase and after studying the feasibility of the constructed ontology and analyzing knowledge sources, we formally define the ontology as following.

Formal Definition of constructed ontology. The ontology O will be defined in this form: O=(C, R, Hc, Rel, A). C and R are defined as disjoint sets dedicated for concepts and relations respectively. Hc is the hierarchy of concepts which is represented as $Hc \subset C \times C$. The Rel set define the semantic and non-taxonomic relations with two associated functions. It is represented as Rel: $R \rightarrow C \times C$ with the domain is defined as dom: $R \rightarrow C$, dom(R) = $\prod 1$ (Rel (R)) and the range is represented as range: $R \rightarrow C$, range(R) = $\prod 2$ (rel (R)) co-domain.

The instantiation of the formal definition of ontology constructed can be represented as shown in Fig. 1:

This light fragment represented in figure 1 contains the set of concepts ***C= {C1 : lexicalEntry, C2 : form, C3 : form Representation, C4 : representation}***, a set of relations ***R={r1 : hasForm, r2 : hasFormRepresentation}*** and a set of concept hierarchy ***Hc={H1c}.***

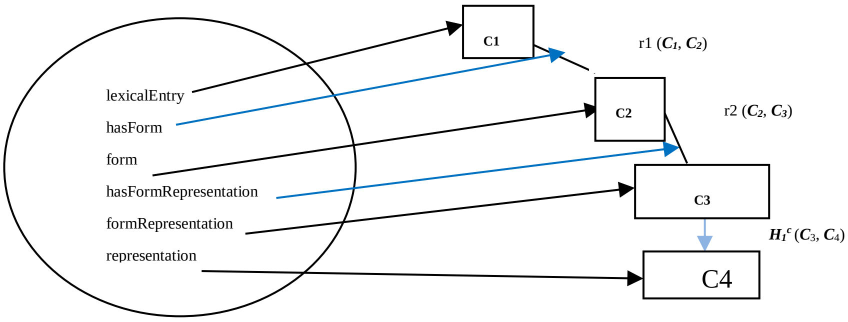


Fig. 1. Representation of LMF core

In order to instantiate the ontology, we have to define the KB[4] set as: ***KB= (O, I, Inst, Instr):*** With ***O = (C, R, HC, Rel, A)*** is the LMF ontology defines formally above, I is the instances set, $inst : C \to 2I$ translated the instantiation function of concepts and $instr : R \to 2\ I \times I$ is the instantiation function of relation.

4 Ontology construction

The ontology construction is an important step in the whole process of mapping LMF-OWL. The target for the development of ontological version is the establishment of an interoperable environment enabling the management of lexical resources. For this reason, construction such framework requires the presence of applications that can make possible the exchange of un-formal and unstructured data across the web.

Thus, the prototype is described in five steps: LMF Entities, Namespaces, LMF classes, LMF subclasses and properties as follows.

4.1 LMF Entities

LMF entities are considered as assertion in the new ontology. To automate the construction of this task, we need to define a new pattern following the JAPE[5] syntax:

```
Pattern1: ({Entity.name=="lmf"}) : NewOWLEntiy
- - > : NewOWLEntiy = {value= "http://www.lexicalmarkupframework.org#", rule = R1p}
```

When defining a new entity named "lmf", a new one has to be defined in OWL. This entity will have as value = "http: //www.lexicalmarkupframework.org#." The set is described under the name NewOWLEntity, and therefore, a specific process must be associated as well.

[4] KB : Knowledge Base

[5] JAPE : Java Annotation Pattern Engine

4.2 Namespaces

Automating the construction of namespaces in the OWL-DL variant requires the definition of a new pattern using the same syntax JAPE:

```
Pattern 2: ({Input.var=="URI"}) : NewOWLUsedNamespace
- - > : NewOWLUsedNamespace = {xmlns = "URI#", rule = R2p}
```

The definition of a new namespace is designed by the pattern presented. The new namespace will have as value = "# URI." The set is described under the name NewOWLUsedNamespace.

4.3 Entities and LMF classes

A set of assertions have to be made after the namespaces accuracy. This set enriches the output with comments, labels, version etc. Automating the construction of OWL-DL headers requires the following pattern:

```
Pattern 3: ({Input.var1=="URI"}{Input.var2=="label"}) : NewOWLHeader
- - > : NewOWLHeader = {element = "owl:Ontology", rule = R3p}
```

The pattern defined has as an entry, an URI and a label and provides as an output a new OWL element noted owl: Ontology.

OWL classes are considered as basic components in the resulted ontology. All these classes will be members of the class Thing.

```
Pattern 4: ({Input.var1=="URI"}{Input.var2=="class"} {"class" ∈ LMFClasses}) :
NewOWLClass
- - > : NewOWLClass = {element = "owl:Class", rule = R4p}
```

The pattern requires an URI and the class name to produce a new element noted owl: Class. However, one condition must be fulfilled before developing any process. It is mandatory to verify the belonging of the new class to the set of LMFClasses defined in LMF and conceived in the previous section.

4.4 LMF SubClasses

The restrictions list may also contain sub-classes. The subclasses construction mechanism conceived in OWL-DL is defined using the following new pattern:

```
Pattern 5: ({Input.var1=="Entity"}
{Input.var2=="Class"}{var2 ∈ LMFClasses}
{Input.var3=="SubClass"}{var3 ∈ LMFSubC}) : NewOWLSubClass
- - > : NewOWLSubClass = {element = "rdfs:subClassOf", rule = R5p}
```

LMF subClass definition is represented by the pattern defined. The Subclass requires three variables: the entity, the subclass and the class to which it belongs. Two conditions must be fulfilled: the class value of variable 2 and the subclass value of the variable 3 must belong respectively to class and subclass set predefined in LMF. This pattern will be described under the name of *NewOWLSubClass*.

4.5 LMF Properties

Many informations are interpreted as attributes in either the core or extensions of LMF packages.

```
Pattern 6: ({Input.var1=="Entity"}
{Input.var2=="Property"}{var2 ∈ LMFProperties}
{Input.var3=="range"}
{Input.var4=="domain"}
{Input.var5=="rangeEntity"}) : NewOWLProperty
- - > : NewOWLProperty = {element = "owl:datatypeProperty", rule = R6p}
```

The accuracy of a LMF property is represented by the previous pattern. The property requires three key variables which are "property", "range" and "domain", other ones are optional.

5 General architecture of the method

After conceiving the method for the automatic construction of ontology, we have now to describe the implementation of the complete method to validate our prototype. In this section, we will describe the general architecture of our prototype.

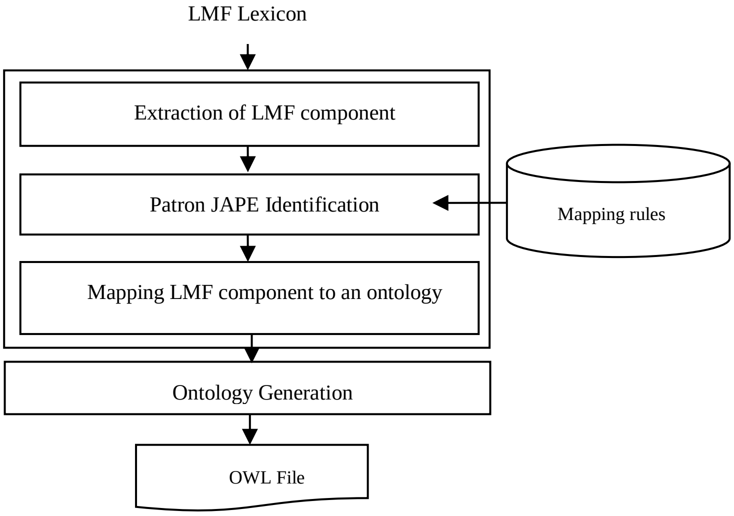

LMF Lexicon
Extraction of LMF component
Patron JAPE Identification
Mapping rules
Mapping LMF component to an ontology
Ontology Generation
OWL File

Fig. 2. General Architecture of the methods

Mapping process input is a serialized LMF lexicon into an XML file which may contains all LMF components or simply restraint set of these constituents. In further section we will explain different methods used to extract the XML component, identification of suitable JAPE pattern and the mapping process using the appropriate rules.

5.1 Extraction of LMF component

The first step of the mapping process is to extract the LMF constituent represented by an XML tag in the LMF lexicon. This is an important step in the mapping process. Figure 3 illustrates this module.

```
   <?xml version="1.0" encoding="UTF-8"?>
<lexicalRessource dtdVersion="16">
  <globalInformation>
    <feat att="languageCoding" val="ISO 639-3"/>
    <feat att="scriptCoding" val="ISO 15 924"/>
  </globalInformation>
  <lexicon>
    <feat att="language" val="arab"/>
    <lexicalEntry morphologicalPatterns="intransitifVerb">
      <feat att="partOfSpeech" val="verb"/>
      <feat att="root" val="ج_ل_س"/>
      <feat att="scheme" val="فَعَلَ"/>
      <lemma>
        <feat att="writtenForm" val="جَلَسَ"/>
        <feat att="writtenForm" val="-"/>
        <feat att="type" val="صحيح"/>
      </lemma>
    </lexicalEntry>
  </lexicon>
</lexicalRessource>
```

Lexical Resource

Fig. 3. Extraction of LMF component

The example in figure 3 concerns the verb jalasa جلس (to sit). The different characteristics related to the verb are compliant to the LMF standard (ISO-24613).

5.2 Identification of the JAPE pattern

The development of the JAPE pattern identification requires a set of tests in order to verify the coincidence of LMF extracted component with the selected JAPE pattern. Figure 4 illustrates all these tests.

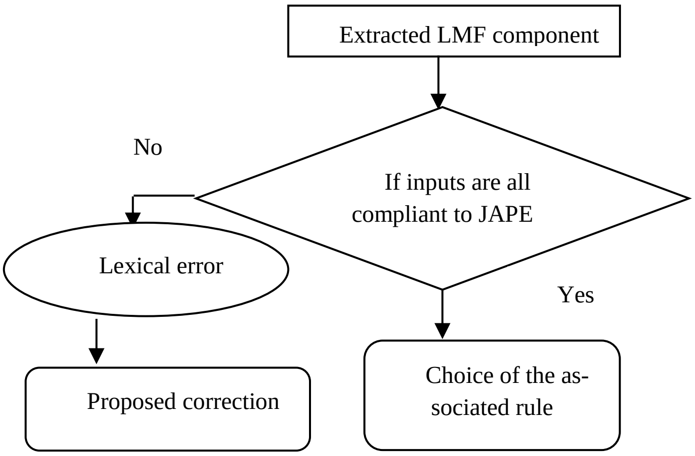

Extracted LMF component
No
If inputs are all compliant to JAPE
Lexical error
Yes
Proposed correction
Choice of the associated rule

Fig. 4. JAPE pattern identification Schema

Figure 4 indicates that the process of identifying the JAPE pattern consists first on verifying compliance in number and value of the component extracted with inputs from each pattern. This verification ends either by the choice of the associated rule or by a proposed correction of the lexical error.

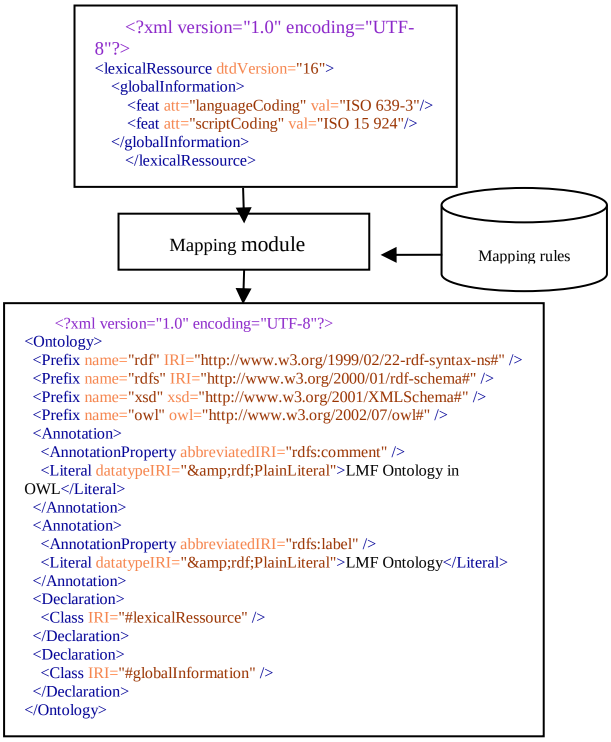
5.3 Mapping from LMF component to the new ontology

The mapping module of LMF component to an ontology using the JAPE pattern is a key phase in the general architecture. This module consists on creating the suitable concept or relation by applying the associated rule. Figure 5 illustrates this mapping.

Figure 5 indicates that the mapping process of the previous fragment detected the presence of the entities (rdf, rdfs, xsd and owl), a label, a commentary and the two concepts: lexicalRessource and globalInformation.

6 Discussion

The evaluation of the OWL variant construction of the LMF process can be measured using the instantiation of the OWL-DL ontology already constructed. Thus, in this section we choose the morphological package extension. This choice is explained by the fact that this extension is considered crucial in most NLP and Semantic Web applications. This extension is described by two different ways in LMF. The first exposed explicitly inflected forms. The second uses flexion paradigms in order to generate various forms derived from the lexical entry. The following built prototype instantiation represents inflectional description of the verb "أكل" (to eat).



```
<?xml version="1.0" encoding="UTF-
8"?>
<lexicalRessource dtdVersion="16">
  <globalInformation>
    <feat att="languageCoding" val="ISO 639-3"/>
    <feat att="scriptCoding" val="ISO 15 924"/>
  </globalInformation>
    </lexicalRessource>
```

```
    <?xml version="1.0" encoding="UTF-8"?>
<Ontology>
 <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
 <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
 <Prefix name="xsd" xsd="http://www.w3.org/2001/XMLSchema#" />
 <Prefix name="owl" owl="http://www.w3.org/2002/07/owl#" />
 <Annotation>
  <AnnotationProperty abbreviatedIRI="rdfs:comment" />
  <Literal datatypeIRI="&amp;rdf;PlainLiteral">LMF Ontology in
OWL</Literal>
 </Annotation>
 <Annotation>
  <AnnotationProperty abbreviatedIRI="rdfs:label" />
  <Literal datatypeIRI="&amp;rdf;PlainLiteral">LMF Ontology</Literal>
 </Annotation>
 <Declaration>
  <Class IRI="#lexicalRessource" />
 </Declaration>
 <Declaration>
  <Class IRI="#globalInformation" />
 </Declaration>
</Ontology>
```

Fig. 5. LMF fragment Mapping

The above example reveals one possible inflected form from a set of 56 possible inflected forms that might have a verb.

Table 1. LMF components Evaluation

JAPE patterns / Number of lexica	LMF Entities	Namespaces	LMF Classes	LMF subclasses	Properties
170	170	680	10710	510	3400

We have applied the constructed prototype for a set of 170 Arabic lexica. The choice of the Arabic language is justified by the availability of these XML files, the necessity of our team work and finally to improve researches in this language.

7 Conclusion

We examined the structure of the LMF standard in order to conceive the OWL-DL ontology core. This ontology may be used for example in a morphosyntactic annotation application. This annotation will play the role of a GATE plugin allowing the disposition of LMF serialization. The underlying idea is to create an interoperable environment evoking dynamism between standards and guidelines. The condition must fulfill these environments is the internal consistency without neglecting the update of modeling involved standards and their serialization. We have proposed a rigorous method based on LMF-JAPE pattern to manage the projection from LMF to OWL format as a serious attempt to succeed interoperability between existing data formats. This method can be used in advance for operating the transitivity between different existing data formats. Consequently, the mapping between LTAG-OWL and HPSG-OWL will be easily operational.

References

1. Assem, V.M., Gangemi, A., and Schreiber, G., 2006, *Conversion of WordNet to a standard RDF/OWL representation*, Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), Genoa, Italy.
2. Berners-Lee, T., Hendler, J., Lassila, O., 2001, *The semantic web*, Scientific american 284.5, 28-37.
3. Bourigault, D., 2002, UPERY: un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus, TALN, Nancy.
4. Buitelaar P., Cimiano P., Magnini B., 2005, Ontology Learning From Text: Methods, Evaluation and Applications, IOS Press.
5. Francopoulo, G., 2013, *Lexical Markup Framework*, US, Great Britain and the United States: ISTE Ltd and John Wiley & Sons, Inc.
6. Grcar M., Klein E., Novak B., 2007, *Using Term-Matching Algorithms for the Annotation of Geo-services*, Postproceedings of the ECML-PKDD Workshops, Springer, Berlin − Heidelberg − New York.Boston, MA: Kluwer Academic Plubisher.
7. Grefenstette, G., 1994, *Explorations in Automatic Thesaurus Discovery*, MA: Kluwer Academic Plubisher, Boston.
8. Haddar, K., Fehri, H., Romary, L., 2012, A prototype for projecting HPSG syntactic lexica towards LMF, JLCL.
9. Ide, N., Romary, L.: A Registry of Standard Data Categories for Linguistic Annotation. In Proc. 4th International Conference on Language Resources and Evaluation - LREC'04 135–138 - http://hal.inria.fr/inria-00099858 (2004)
10. Lhioui, M., Haddar, K., Romary, L., 2014, *Towards modelling LMF lexicons compliant LMF in OWL-DL*, TKE, Berlin.
11. Loukil, N., Ktari, R., Haddar, K., Benhamadou, A., 2010, A normalized syntactic lexicon for arabic verbs and its evaluation within the LKB platform, ACSE, Egypt.
12. NEON Project, D2.5.1 :, 2008, A
13. Library of Ontology design Patterns : reusable solutions for collaborative design of networked ontologies.

14. Sperberg-McQeen, C.M., Burnard, L., 2014, *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, Text Encoding Initiative Consortium Charlottesville, Virginia: the TEI Consortium.
15. TAUS, Report on a TAUS research about translation interoperability, 25 February, 2011.
16. Weissenbacher, D., Nazarenko, A., 2007, Identifier les pronoms anaphoriques et trouver leurs antécédents : l'intérêt de la classification bayésienne, TALN, Toulouse.
17. Wilcock, G., 2007, *An OWL ontology for HPSG*, ACL, Finland.

Word Sense Induction for Better Lexical Choice

Neha Prabhugaonkar[1], Jyoti Pawar[1] and Pushpak Bhattacharyya[2]

[1] Department of Computer Science and Technology,
Goa University, Goa
nehapgaonkar.1920@gmail.com, jyotidpawar@gmail.com
[2] Department of Computer Science and Engineering,
IIT Bombay, Powai
pb@cse.iitb.ac.in

Abstract. Most words in natural languages are *polysemous* in nature that is they have multiple possible meanings or *senses*. The sense in which the word is used determines the translation of the word. We show that incorporating a sense-based translation model into statistical machine translation model consistently improves translation quality across all different test sets of five different language-pairs, according to all eight most commonly used evaluation metrics. This paper is an investigation on how to initiate research in word sense disambiguation and statistical machine translation for under-resourced languages by applying *Word Sense Induction*.

1 Introduction

Word Sense Disambiguation or WSD is the ability to identify the meaning of words in context in a computational manner [1]. A wide variety of approaches ranging from supervised to unsupervised algorithms have been proposed. Supervised approaches ([2] and [3]) which rely on sense annotated corpora have proven to be more successful, and they substantially outperform knowledge-based and unsupervised approaches ([4] and [5]). However, creation of sense annotated corpora is always costly and time-consuming, especially for the resource scarce languages.

1.1 Use of WSD models in SMT

WSD is often assumed to be an intermediate task, which should then help higher level applications such as Machine Translation or Information Retrieval. However, WSD is usually performed and evaluated as a standalone task but there have been very few efforts to integrate the learned WSD models into full SMT systems. Some of the reasons are:

- Most of the WSD approaches assign senses with the aid of dictionaries, or other lexical resources such as WordNet; it is difficult to adapt them to new domains or to languages where such resources are scarce.

pp. 35–44; rec. 2015-01-25; acc. 2015-02-27

- A related problem concerns the granularity of the sense distinctions which is fixed, and may not be entirely suitable for different applications [6].
- There is a risk that an important sense will be missed, or an irrelevant sense will influence the results.
- In many cases, lexical resources like WordNet is very precise, defining senses which are similar and hard to distinguish.

1.2 Why WSI for SMT?

Initially, WSD was mainly applied and developed on English texts, because of the broad availability and the prevalence of lexical resources compared to other languages. Due to the lack of availability of large lexical resources i.e. sense inventories (dictionaries, lexical databases, WordNets, etc.) and parallel sense-tagged corpora it is difficult to start working on WSD for under-resourced languages (Tamil, Konkani, Telugu, etc.). To account for under-resourced languages, one can easily adopt techniques aimed at the automatic discovery of word senses from text, a task called Word Sense Induction.

Word Sense Induction (WSI) is a task of automatically inducing the underlying senses of word tokens given the surrounding contexts where the word tokens occur. The biggest difference from word sense disambiguation lies in that WSI does not rely on a predefined sense inventory.

Recent work in Machine Translation ([7] and [8]) and Information Retrieval [9] indicates that induced senses can lead to substantial improvement in performance where methods based on a fixed sense inventory such as HowNet have previously failed ([10] and [11]). Therefore, We adopt the similar approach of Xiong and Zhang [8] by resorting to Word Sense Induction (WSI) that is related to but different from WSD.

The advantages of using WSI are:

- It actually performs word sense disambiguation.
- Aims to divide the occurrence of a word into a number of classes.
- Makes objective evaluation easy if it is domain-specific.

The rest of the paper is structured as follows: Section 2 describes the Related work. In Section 3, we describe the SMT system and its essential components. In Section 4, we provide details about the experiments conducted and results obtained. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Standard WSD for SMT

Carpuat and Wu [10] integrated the translation predictions from a state-of-the-art Chinese WSD system [12] into a Chinese-English word-based SMT system using the ISI ReWrite decoder [13]. They used the WSD model predictions either *to substitute for translation candidates of their translation model* or *to post edit the output of their SMT system.* The authors reported that WSD does not yield significantly better translation quality than the SMT system alone.

2.2 Redefined WSD for SMT

Vickrey et al., [7], redefined the standard WSD problem for SMT as a *word translation task* - predicting possible target translations rather than senses for ambiguous source words. The translation choices for a word w were defined as the set of words or phrases aligned to w, as gathered from a word-aligned parallel corpus. The authors reported that they were able to improve their models accuracy on a simplified word translation task.

Chan et al., [14], successfully integrated a state-of-the-art WSD system into a state-of-the-art Hierarchical phrase-based system, Hiero [15]. They introduced two WSD-related additional features into the log-linear model of SMT. Carpuat and Wu [10] also used the redefined WSD for SMT and further adapted it for multi-word phrasal disambiguation. They both reported that redefined WSD system improves the performance of a state-of-the-art SMT system on actual translation task.

Although the redefined WSD has proved helpful for SMT, recently, Xiong and Zhang [8] re-investigated the question of whether pure senses are useful for SMT by using WSI. They proposed a sense-based translation model to integrate word senses into SMT which enables the decoder to select appropriate translations for the source words according to the inferred senses for these words using Maximum Entropy classifiers. The authors reported that the proposed model substantially outperforms not only baseline but also the previously redefined WSD.

3 The SMT system

To build a representative baseline SMT system, we restricted ourselves to making use of freely available tools. Since our focus is not on a specific SMT architecture, we used the `cdec`[3] [16] toolkit trained in a standard fashion for our experiments. The detailed architecture of the SMT system is shown in Figure 1.

3.1 Data Preprocessing

We preprocess the source side of our bilingual training data as well as development and test set by removing stop words and rare words. From the preprocessed training data, we extract all possible pseudo documents for each source word type. The collection of these extracted pseudo documents is used as a corpus to train a HDP-based WSI model for the source word type. In this way, we can train as many HDP-based WSI models as the number of word types kept after preprocessing.

3.2 Sense Annotation

To obtain word senses for any source words, we build a sense tagger that relies on the nonparametric Bayesian model based word sense induction ([17], [18]) similar to Xiong and Zhang [8]. We used HDP-based WSI[4] [19] to predict sense

[3] `http://www.cdec-decoder.org/`

[4] `http://www.cs.cmu.edu/\~chongw/resource.html`

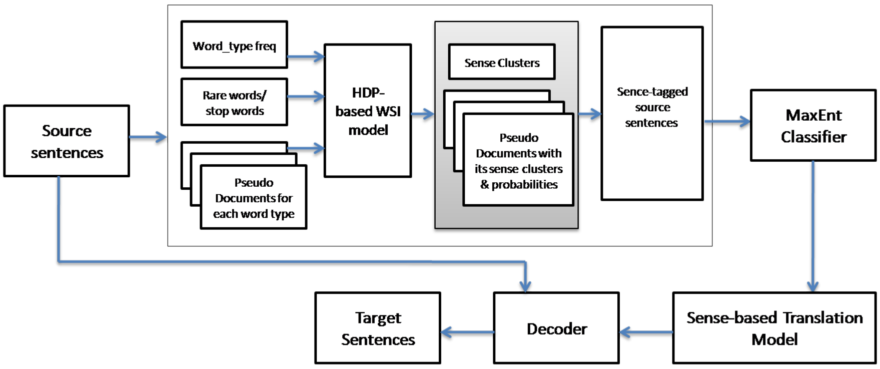

Source sentences
Word_type freq
Rare words/ stop words
Pseudo Documents for each word type
HDP-based WSI model
Sense Clusters
Pseudo Documents with its sense clusters & probabilities
Sence-tagged source sentences
MaxEnt Classifier
Sense-based Translation Model
Decoder
Target Sentences

Fig. 1. Architecture of SMT system

clusters and to annotate source words in our training/development/test sets with these sense clusters. We individually build a HDP-based WSI model per word type and train these models on the training data. The sense for a word token is defined as the most probable sense according to the per-document sense distribution estimated for the corresponding pseudo document that represents the surrounding context of the word token.

3.3 Alignment Model

The alignment model was trained with `fast-align` alignment tool which is a variant of the aligner proposed by Dyer et al., [20]. The alignment algorithm is trained in either direction and are symmetrized using grow-diag-final heuristics.

3.4 Language Model

The Hindi language model is a five gram model trained on the **Hindi** side of the parallel corpora using a publicly available software, the KenLM[5] [21] toolkit. We used additional monolingual corpora[6] [22] of ≈45 million lines and included more Hindi monolingual corpora[7] for language model training.

3.5 Sense-based Translation Model

The sense-based translation model estimates the probability that a source word c is translated into a target phrase e given contextual information, i.e. word senses that are obtained using the HDP-based WSI. We adopt the same approach of Xiong and Zhang [8] to build the sense-based Translation Model.

[5] `https://kheafield.com/code/kenlm/`

[6] `https://lindat.mff.cuni.cz/repository/xmlui/browse?value=hin\&type=language`

[7] `http://www.cfilt.iitb.ac.in/wsd/annotated\_corpus/`

Table 1. WSI-based SMT improves BLEU, GTM-3, NIST, WER, PER, TER, METEOR and ROUGE-L across all language-pair datasets.

Lang-pair	Expts	BLEU	GTM-3	NIST	WER	PER	TER	METEOR	ROUGE
Eng-Hin	SMT	0.2619	0.3253	5.8787	0.649	0.5155	0.6346	0.2581	0.0727
	WSI+SMT	0.2747	0.3394	6.1792	0.62	0.4873	0.6021	0.2665	0.0771
Ben-Hin	SMT	0.3674	0.4063	7.4327	0.4424	0.3918	0.4537	0.315	0.1008
	WSI+SMT	0.3761	0.4151	7.5075	0.4347	0.3856	0.4479	0.3184	0.1004
Mar-Hin	SMT	0.4096	0.4231	7.8353	0.4211	0.3866	0.4265	0.3335	0.1172
	WSI+SMT	0.4156	0.4319	7.5009	0.4581	0.4262	0.4475	0.3526	0.118
Tam-Hin	SMT	0.2057	0.2386	5.1119	0.671	0.5334	0.6544	0.223	0.0967
	WSI+SMT	0.2157	0.2529	4.8329	0.7222	0.5843	0.7067	0.2386	0.096
Tel-Hin	SMT	0.2822	0.3415	5.8713	0.606	0.556	0.592	0.288	0.1215
	WSI+SMT	0.2976	0.3556	6.4549	0.5088	0.4666	0.5208	0.273	0.1218

4 Experimental Details

4.1 Datasets and Resources Used

We used five different language pairs in our experiments - representing a wide range of diversities, such as language family (Indo-Aryan: Hindi, Bengali and Marathi, Dravidian: Tamil and Telugu and West Germanic: English), languages with high structural divergence and morphological manifestations (English is structurally classified as a Subject-Verb-Object (SVO) language with poor morphology whereas Hindi is a morphologically rich, Subject-Object-Verb (SOV) language), etc. The target language for all the languages is **Hindi**.

The datasets belonged to the tourism and health domains (25,000+25,000 sentences) from the ILCI corpora. We normalized the corpus to solve issues related to incorrect characters, redundant Unicode representation of some Indic characters, etc. The English corpus was tokenized using the Stanford tokenizer[8] [23] and for Indian languages, we used NLP Indic Library[9] [24].

For every language pair, the corpus was split up as follows: training set of 48000 sentences, development test set of 1000 sentences and test set of 1000 sentences. The training, development test and test splits are completely parallel across the five language-pairs involved.

[8] `http://nlp.stanford.edu/software/`

[9] `https://github.com/anoopkunchukuttan/indic\_nlp\_library`

Table 2. Examples of translations drawn from the English-Hindi test set.

Example 1	Input	in Delhi many types of food of India and abroad are served
	Sense-based SMT output / Reference	दिल्ली में भारत और विदेशों के अनेक प्रकार के भोजन परोसे जाते है ।
Example 2	Input	this medicine is mainly used for ulcer , asthma and bronchitis.
	Sense-based SMT output / Reference	इस औषधि का विशेष रूप से प्रयोग अल्सर, दमा और ब्रोंकाटिस के लिये किया जाता है ।
Example 3	Input	the journey of namdapha is easy and also inexpensive .
	Baseline	नमदफा का **सफर** भी आसान और सस्ता है ।
	Sense-based SMT output / Reference	नमदफा की **यात्रा** आसान है और सस्ती भी ।
Example 4	Input	along with sunrise the stir of the devotees start at the ramghat .
	Baseline	सूर्योदय के साथ ही रामघाट पर स्रद्धालुओ की हलचल **शुरू** हो जाता है ।
	Sense-based SMT output / Reference	सूर्योदय के साथ ही रामघाट पे स्रद्धालुओ की हलचल **आरंभ** हो जाती है
Example 5	Input	shampoo a little while after the massage .
	Baseline	मसाज के थोडी देर बाद शेम्पू ।
	Sense-based SMT output / Reference	मसाज के थोडी देर बाद शेम्पू **कर ले** ।

Table 3. Number of translations which exactly matched with the reference sentences.

Language-pair	**Baseline**	**WSI-based**	**Overlap between**
	SMT	**SMT**	**Baseline & WSI-based SMT**
English-Hindi	39	44	11
Bengali-Hindi	63	66	23
Marathi-Hindi	57	59	38
Tamil-Hindi	16	19	9
Telugu-Hindi	29	29	19

4.2 Results and Analysis

As mentioned, our experiments were on Indian language (Bengali, Marathi, Tamil, Telugu) to Hindi translation and English to Hindi translation. To measure the impact of using sense-based Translation Model on translation quality, we used the most commonly used automatic evaluation metrics to evaluate the translations obtained. Apart from the widely used BLEU [25] and NIST [26], we also evaluate translation quality with METEOR [27] without using WordNet synonyms to match translation candidates and references, General Text Matcher (GTM-3), Word Error Rate (WER), Position-independent word Error Rate (PER), Translation Edit Rate (TER) [28] and ROUGE. These metrics have proved to relate well with both adequacy and fluency. The results are shown in Table 1.

Using sense-based Translation Model in SMT yields better translation quality on all language-pair test sets, as measured by all eight commonly used automatic evaluation metrics.

Table 2 show examples of translations drawn from the English-Hindi test set. Analysis says that WSI-based translation model helps decoder to give better rankings and lexical choices then the baseline translation probabilities (see Example 3 and 4). Examples 1-5 are the translations which exactly matched with reference sentences. We came across many such examples where the lexical item proposed by the WSI-based translation model was better than the baseline system which resulted in increase in performance of the MT system.

5 Conclusion and Future Work

We have shown that sense-based Translation Model improves the translation performance of an Indian language SMT system and its improvement is statistically significant in terms of all eight evaluation metrics. Word senses induced automatically by the HDP-based WSI are very useful for Machine Translation for under-resourced languages. The sense-based Translation Model in SMT is effective at choosing the correct and appropriate lexical choice for an ambiguous word.

Our future work will be to build a sense-based Hindi language model by inducing sense clusters for words in the target language. We would also like to explore whether integrating learned WSD Model in SMT for same Indian language-pairs improves translation quality or not and perform a comparative study.

References

1. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. **41** (2009) 10:1–10:69
2. Ng, H.T., Lee, H.B.: Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In: Proceedings of the 34th Annual Meeting

on Association for Computational Linguistics. ACL '96, Stroudsburg, PA, USA, Association for Computational Linguistics (1996) 40–47

3. Lee, Y., Ng, H., Chia, T.: Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. In: Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. (2004) 137–140
4. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation. SIGDOC '86, New York, NY, USA, ACM (1986) 24–26
5. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 411–418
6. Brody, S., Lapata, M.: Bayesian word sense induction. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. EACL '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 103–111
7. Vickrey, D., Biewald, L., Teyssier, M., Koller, D.: Word-sense disambiguation for machine translation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), Vancouver, Canada (2005)
8. Xiong, D., Zhang, M.: A sense-based translation model for statistical machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, Association for Computational Linguistics (2014) 1459–1469
9. Veronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech and Language **18** (2004) 223–252
10. Carpuat, M., Wu, D.: Word sense disambiguation vs. statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 387–394
11. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '93, New York, NY, USA, ACM (1993) 171–180
12. Wu, D., Su, W., Carpuat, M.: A kernel pca method for superior word sense disambiguation. In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics. ACL '04, Stroudsburg, PA, USA, Association for Computational Linguistics (2004)
13. Germann, U.: Greedy decoding for statistical machine translation in almost linear time. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. NAACL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 1–8
14. Chan, Y.S., Ng, H.T., Chiang, D.: Word sense disambiguation improves statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, Association for Computational Linguistics (2007) 33–40

15. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 263–270
16. Dyer, C., Weese, J., Setiawan, H., Lopez, A., Ture, F., Eidelman, V., Ganitkevitch, J., Blunsom, P., Resnik, P.: Cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In: Proceedings of the ACL 2010 System Demonstrations. ACLDemos '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 7–12
17. Yao, X., Durme, B.V.: Nonparametric bayesian word sense induction. In: Graph-based Methods for Natural Language Processing, The Association for Computer Linguistics (2011) 10–14
18. Lau, J.H., Cook, P., McCarthy, D., Newman, D., Baldwin, T.: Word sense induction for novel sense detection. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. EACL '12, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 591–601
19. Wang, C., Blei, D.M.: A split-merge mcmc algorithm for the hierarchical dirichlet process. CoRR **abs/1201.1657** (2012)
20. Dyer, C., Chahuneau, V., Smith, A.N.: A simple, fast, and effective reparameterization of ibm model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2013) 644–648
21. Heafield, K.: Kenlm: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. WMT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 187–197
22. Bojar, O., Diatka, V., Rychly, P., Stranak, P., Suchomel, V., Tamchyna, A., Zeman, D.: Hindencorp - hindi-english and hindi-only corpus for machine translation. In Chair), N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (2014)
23. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. ACL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 423–430
24. Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R.M., Bhattacharyya, P.: Shata-anuvadak: Tackling multiway translation of indian languages. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014. (2014) 1781–1787
25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 311–318
26. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research. HLT '02, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2002) 138–145
27. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or

Summarization, Ann Arbor, Michigan, Association for Computational Linguistics (2005) 65–72

28. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: In Proceedings of Association for Machine Translation in the Americas. (2006) 223–231

Annotating Geographical Entities

Alexandru Sălăvăstru, Daniela Gîfu

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
{alexandru.salavastru,daniela.gifu}@info.uaic.ro

Abstract. This paper describes a study based on exploration of relations between geographical entities. We suggested a new tool for training and evaluation required by related annotation experiments. It relates to an annotator used for semi-automatic annotation, starting with the geography manual. We define fifteen types of entities: *location*, *geo_position*, *geology, landform, clime, water, dimension, person, organization*, *URL*, *Timex*, *resource*, *industry, cultural, unknown* with their specific subtypes. Moreover, we present the annotation conventions for three semantic relations: *referential*, *structural* and *spatial*, considered to be optimal operators in understanding a geographical manual. A part of the annotation is done manually, while the other part is done automatically, such as the token, lemma, part-of-speech. The study is intended to create a tool for the automatic detection of semantic relations in texts on geographic issues such as geography manuals, travel guides, geography atlases, etc., in order to help children, professors, guides, PR specialists and to be useful for tourists, generally to discover the complexity and the beauty of the nature.

Keywords: geography manual, entities, annotation conventions, semantic relation, annotator

1 Introduction

Starting with the NAACL 2003 Workshop on the Analysis of Geographic References[1], a new community of NLP researchers and engineers focused on different aspects of the geographic text analysis task. The motivation for our study relies on the need for objectivity in the interpretation of semantic relations between geographical entities. We present a new tool, called RelAnn (*Relations Annotator*) used for annotation and semi-automatic extraction of entities and for manual annotation of semantic relations on corpora on geographical topics. We establish annotation conventions for the semantic relations that will be the ground of defining patterns which contain information at lexical and syntactic level for the automatic recognition of those relations from all kinds of geographical texts (geography manuals, tourist guides, atlases) and to extract their functionality (e.g. natural resources from Apuseni Mountains).

Our corpus is a Romanian Geography manual containing about 160 pages of geographical entities and semantic relations between two arguments (entities). Moreover,

[1] http://www.kornai.com/NAACL/WS9/orig.html

we create a Gold Corpus for Romanian language based on geographical issues. The annotation process was a long one, preceded by the following modules: POS tagger [17], NP-Chunking [18], NER (*Name Entity Recognizer*) [5] and RARE (*Robust Anaphora Resolution Engine*) [6].

The paper is structured as follows: Section 2 briefly describes the background related to annotating entities and semantic relations, Section 3 discusses the annotation process using the RelAnn tool, Section 4 presents some results and statistics interpretation and finally, Section 5 depicts some conclusions and directions for future work.

2 Background

RelAnn is an annotator inspired by RECON [9] for automatic entity recognition. ANNIE tool, included in GATE [4] is well-known for this task. Until now, we used PALinkA[2] for annotating corpora in several similar projects, for purposes including: anaphoric and co-referential links in a parallel French-English corpus, summarization, different versions of the Centering Theory, co-references in email messages and web pages, or for Romanian entities. A relevant example is the annotation of the "Quo Vadis" novel [3].

The most common annotation tools are the web-based ones. One of them, called BRAT, is an annotation tool used in many research papers, which aims at the extraction of biomedical events [7], co-references in scientific papers [14], entity annotation on medical corpus [15], etc.

Another tool, WORDFREAK [11], is used in active-learning for human correction of automatically annotated data. Others like CAT, CELCT Annotation Tool[3] [1], suggest a multi-layer annotation concept, and give annotators the ability to create labels and relations with multiple properties such as font size, color, type of relation, and so on.

In order to avoid issues such as local storage, cross-platform deployment, we found Anaphora tool [2], where annotations are divided into *Entity* and *Relation* types, both stored in XML files. This methodology is similar to what we suggest in our study. Similar to Anaphora is eHost/ChartReader [19] made for multiple annotators sharing the same text, but also relies on a remote installation of ChartReader. It offers only basic relations between annotations (primarily for co-reference) and stores them in XML files.

3 Annotation process. Tools and conventions

In this section we will describe the entire annotation process including the annotation tool corpora and the annotation methodology, with many similarities found in SpaceML[10].

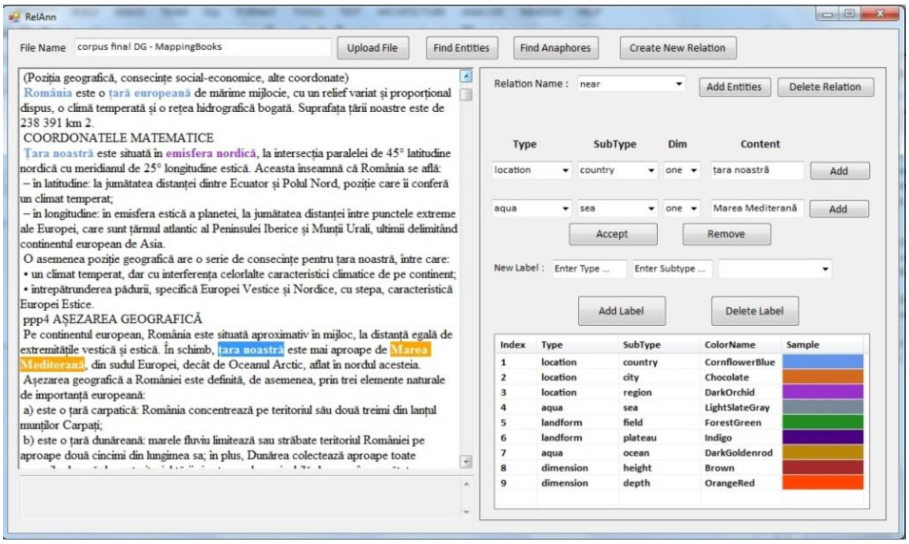
[2] PALinkA was created by Constantin Orăşan in the Research Group in Computational Linguistics, at the School of Law, Social Sciences and Communications, Wolverhampton.

[3] CELCT Annotation Tool developed by Centre for the Evaluation of Language and Communication Technologies (CELCT) - http://www.celct.it/.

3.1 RelAnn Tool

As mentioned before, RelAnn (Fig. 1.) was inspired by the RECON tool, but there are major differences between them. RECON gives the possibility to create n-ary relations and marks-up text spans without any relation definition. It allows annotators to create long-chain co-reference with import / export for XML standoff format. RelAnn is a new semi-automatic tool that creates and annotates entities and semantic relations. The main course was to create a user-friendly, easy-to-use application, but also with fewer restrictions as possible and so far it seems very efficient in doing our tasks.

First, we check the recognized entities. Then we define relations with argument range and add types of entities. Our tool is not restrictive to predefined features, but gives the possibility to create any kind of relations and add different types of entities with their particular color. At each step, we identify in the text entities that have semantic relations between them, a trigger that signals the relation, a direction from one argument to another and save them as a RELATION entry in the XML file. This tool can be used to annotate entities from different domains, such as: *biology*, *computer science*, *literature*, *astronomy*, *physics* and so on. Another important feature is that each text has relations and entities stored separately, and after upload, it shows your progress on that particular file.



File Name corpus final DG - MappingBooks

(Poziția geografică, consecințe social-economice, alte coordonate)
România este o țară europeană de mărime mijlocie, cu un relief variat și proporțional dispus, o climă temperată și o rețea hidrografică bogată. Suprafața țării noastre este de 238 391 km 2.
COORDONATELE MATEMATICE
Țara noastră este situată în emisfera nordică, la intersecția paralelei de 45° latitudine nordică cu meridianul de 25° longitudine estică. Aceasta înseamnă că România se află:
– în latitudine: la jumătatea distanței dintre Ecuator și Polul Nord, poziție care îi conferă un climat temperat;
– în longitudine: în emisfera estică a planetei, la jumătatea distanței între punctele extreme ale Europei, care sunt țărmul atlantic al Peninsulei Iberice și Munții Urali, ultimii delimitând continentul european de Asia.
O asemenea poziție geografică are o serie de consecințe pentru țara noastră, între care:
• un climat temperat, dar cu interferența celorlalte caracteristici climatice de pe continent;
• întrepătrunderea pădurii, specifică Europei Vestice și Nordice, cu stepa, caracteristică Europei Estice.
ppp4 AȘEZAREA GEOGRAFICĂ
Pe continentul european, România este situată aproximativ în mijloc, la distanță egală de extremitățile vestică și estică. În schimb, țara noastră este mai aproape de Marea Mediterană, din sudul Europei, decât de Oceanul Arctic, aflat în nordul acesteia.
Așezarea geografică a României este definită, de asemenea, prin trei elemente naturale de importanță europeană:
a) este o țară carpatică: România concentrează pe teritoriul său două treimi din lanțul munților Carpați;
b) este o țară dunăreană: marele fluviu limitează sau străbate teritoriul României pe aproape două cincimi din lungimea sa; în plus, Dunărea colectează aproape toate

Relation Name : near

Type	SubType	Dim	Content
location	country	one	țara noastră
aqua	sea	one	Marea Mediterană

New Label : Enter Type ... Enter Subtype ...

Index	Type	SubType	ColorName	Sample
1	location	country	CornflowerBlue	🟦
2	location	city	Chocolate	🟧
3	location	region	DarkOrchid	🟪
4	aqua	sea	LightSlateGray	◼
5	landform	field	ForestGreen	🟩
6	landform	plateau	Indigo	🟪
7	aqua	ocean	DarkGoldenrod	🟨
8	dimension	height	Brown	🟫
9	dimension	depth	OrangeRed	🟥

Fig. 1. RelAnn interface working session

3.2 Corpora

While preparing the preliminary conclusions in the configuration model, we decided to include in our corpora a geography manual [12] containing 160 pages and almost 37.000 tokens. At first, the text was edited in PDF, so we applied the boiling-plate technology to obtain raw text in txt format and then we made corrections to the raw text. We intend to enlarge our corpora with numerous texts from the geographical field and thus improving discovering rates and information extraction patterns.

3.3 Methodology

This work is based on a set of principles for relation inventories found in Vivi Năstase's book [12], which cites Levi [8] and Ó Séaghdha and Copestake [16]. We followed some guidelines such as: inventory relations should give a good coverage, provide useful semantic information, and entity classes should be well defined, with no overlapping.

The research consists in: pre-processing the Corpus; annotating entities; annotating semantic relations; and evaluation.

3.3.1 Pre-processing the Corpus

The Romanian automatic pre-processing chain applied on raw texts of the book consists of the following tasks, executed in sequence:

– Segmentation (splitting the text in sentences).

– Tokenization (demarcates words or word compounds, but also numbers, punctuation marks, abbreviations, etc.).

– Lemmatization (determines lemmas of words).

– Part-of-speech tagging (identifies POS categories and morpho-syntactic information of tokens).

This is a part from the sentence segmentation annotation in XML standoff format:

```
....
<S ID="s9" offsetStart="1441" offsetStop="1492"/>
....
```

The tokenization, lemmatization and POS tagging are realized in one step.

For instance: *România ... o țară europeană* / (EN) *Romania ... an European country*.

```
<W Case="direct" Definiteness="yes" Gender="feminine" ID="w10.134" LEMMA="românie" MSD="Ncfsry" Number="singular" POS="NOUN" Type="common" offsetStart="2256" offsetStop="2263" text="România"/>
...
<W Case="direct" Gender="feminine" ID="w10.136" LEMMA="un" MSD="Tifsr" Number="singular" POS="ARTICLE" Type="indefinite" offsetStart="2269" offsetStop="2270" text="o"/>
<W Case="direct" Definiteness="no" Gender="feminine" ID="w10.137" LEMMA="țară" MSD="Ncfsrn" Number="singular" POS="NOUN" Type="common" offsetStart="2271" offsetStop="2275" text="țară"/>
```

```
<W Case="direct" Definiteness="no" Gender="feminine" ID="w10.138" LEMMA="european" MSD="Afpfsrn" Number="singular" POS="ADJECTIVE" offsetStart="2276" offsetStop="2285" text="europeană"/>
```

– Noun phrase chunking [16] (recognizing the chunks that consist of noun phrases (NPs)). We provide the word POS tagging information to observe the id references between NP chunks and words that form that chunk.

Ex: *Suprafața țării noastre.* / (EN) *Our country area*.

```
<W Case="direct" Definiteness="yes" Gender="feminine" ID="w11.1" LEMMA="suprafață" MSD="Ncfsry" Number="singular" POS="NOUN" Type="common" offsetStart="2400" offsetStop="2409" text="Suprafața"/>
<W Case="oblique" Definiteness="yes" Gender="feminine" ID="w11.2" LEMMA="țară" MSD="Ncfsoy" Number="singular" POS="NOUN" Type="common" offsetStart="2410" offsetStop="2415" text="țării"/>
<W Case="oblique" Gender="feminine" ID="w11.3" LEMMA="meu" MSD="Ds1fsop" Number="singular" POS="DETERMINER" Person="first" Possessor_number="plural" Type="possessive" offsetStart="2416" offsetStop="2423" text="noastre"/>
<NP HEADID="11.1" ID="n106" WORDSID="w11.1, w11.2" offsetStart="2400" offsetStop="2415"/>
<NP HEADID="11.2" ID="n107" WORDSID="w11.2" offsetStart="2410" offsetStop="2415"/>
```

– Named Entity Recognizer (NER – identifies and classifies the entities):

```
<S ID="s10" offsetStart="1493" offsetStop="2399"/>
<W Case="direct" Definiteness="yes" Gender="feminine" ID="w10.134" LEMMA="românie" MSD="Ncfsry" Number="singular" POS="NOUN" Type="common" offsetStart="2256" offsetStop="2263" text="România"/>
<W EXTRA="intranzitiv" ID="w10.135" LEMMA="fi" MSD="Vmip3s" Mood="indicative" Number="singular" POS="VERB" Person="third" Tense="present" Type="predicative" offsetStart="2264" offsetStop="2268" text="este"/>
<W Case="direct" Gender="feminine" ID="w10.136" LEMMA="un" MSD="Tifsr" Number="singular" POS="ARTICLE" Type="indefinite" offsetStart="2269" offsetStop="2270" text="o"/>
<W Case="direct" Definiteness="no" Gender="feminine" ID="w10.137" LEMMA="țară" MSD="Ncfsrn" Number="singular"
```

```
POS="NOUN" Type="common" offsetStart="2271" off-
setStop="2275" text="ţară"/>
<W Case="direct" Definiteness="no" Gender="feminine"
ID="w10.138" LEMMA="european" MSD="Afpfsrn" Num-
ber="singular" POS="ADJECTIVE" offsetStart="2276" off-
setStop="2285" text="europeană"/>
<ENTITY ID="e1" TYPE="location" SUBTYPE="country"
DIMENSION="one" WORDSID="w10.134," offsetStart="2256"
offsetStop="2263" Color="Chocolate" />
<ENTITY ID="e2" TYPE="location" SUBTYPE="country"
DIMENSION="one" WORDSID="w10.137, w10.138" off-
setStart="2271" offsetStop="2285" Color="Chocolate"/>
```

– Anaphora Resolution (RARE - extract co-reference chains). For instance:

România este o țară europeană de mărime mijlocie. Suprafața țării noastre este de 238 391 km ^{2}.

Romania is a medium size European country. Our country area is 238 391 km ^{2}.

3.3.2 Annotating entities

Our intention was to markup entities (15 types and 89 subtypes)[4] as mentioned in a geography manual (see Table 1). Below we mention all entity types with one subtype examples for each. Let's note that we have too many entity subtypes in order to illustrate all.

Table 1. Entity classification with examples

Type	Subtype	Example (RO/EN)
location	county seat	Iași
geo_position	cardinal direction	Est / East
geology	rock	Granite / granit
landform	hill	Dealurile de Vest / Western Hills
clime	climate type	temperate
water	river	Olt / Olt
dimension	latitude	45° N

[4] Let's note that at this moment we have a too large range of entities, the statistical data clearly showing that we have to rethink the classification of entity subtypes.

person	person	Simion Mehedinți
organization	organization	O.N.U
url	-	www.geografie.ro
timex	-	Iulie 2004 / July 2004
resource	coal basin	bazinul Petroșani / basin Petroșani
industry	thermopower	Termocentrala Borzești / Thermal power plant Borzesti
cultural	palace	Peleș
unknown	-	-

3.3.3 Annotating semantic relations

In this paper we introduce three types of semantic relations with their particular subtypes. Each relation holds between two arguments, called *poles*. The part that signals the type of relation, which can be one word or expression, is called *trigger*. Our notations are expressed in XML. We use <S></S> for marking sentences with attributes id and start/stop offset, and <W></W> for words with attributes like ids, lemmas, morphosyntatic analysis, start/stop offset and text. Also, the <NP></NP> annotation for NP chunks with specific ids and references to words that form them and start/stop offset, and <ENTITY></ENTITY> to mark entities with attributes like id, type, subtype, reference to word, colour and start/stop offset. Finally we added to the file the <RELATION></RELATION> with id, type, subtype, from, to and trigger attributes. The XML standoff format is easier to make any modifications to the file without changing the structure.

In the following we give examples for each type of relation, and an XML for one of them.

I. *Referential relations* with two subtypes are listed and illustrated below:

- *coref* (anaphora).
1:[Romania]... 2:[țării noastre]... / (EN) (1:[Romania]... 2:[our country]...) ⇒ [2] coref [1];
- *isa* (a relation from element to class (concept)).
1:[România] este o 2:[țară europeană] / (EN) (1:[Romania] is a 2:[European country]). ⇒ [1] isa [2];

```
<S ID="s10" offsetStart="1493" offsetStop="2399"/>
<W Case="direct" Definiteness="yes" Gender="feminine" ID="w10.134" LEMMA="românie" MSD="Ncfsry" Number="singular" POS="NOUN" Type="common" offsetStart="2256" offsetStop="2263" text="România"/>
<W EXTRA="intranzitiv" ID="w10.135" LEMMA="fi" MSD="Vmip3s" Mood="indicative" Number="singular" POS="VERB" Person="third" Tense="present"
```

```
Type="predicative" offsetStart="2264" offsetStop="2268"
text="este"/>
<W Case="direct" Gender="feminine" ID="w10.136"
LEMMA="un" MSD="Tifsr" Number="singular" POS="ARTICLE"
Type="indefinite" offsetStart="2269" offsetStop="2270"
text="o"/>
<W Case="direct" Definiteness="no" Gender="feminine"
ID="w10.137" LEMMA="țară" MSD="Ncfsrn" Number="singular"
POS="NOUN" Type="common" offsetStart="2271" off-
setStop="2275" text="țară"/>
<W Case="direct" Definiteness="no" Gender="feminine"
ID="w10.138" LEMMA="european" MSD="Afpfsrn" Num-
ber="singular" POS="ADJECTIVE" offsetStart="2276" off-
setStop="2285" text="europeană"/>
<ENTITY ID="e1" TYPE="location" SUBTYPE="country"
DIMENSION="one" WORDSID="w10.134," offsetStart="2256"
offsetStop="2263" Color="Chocolate" />
<ENTITY ID="e2" TYPE="location" SUBTYPE="country"
DIMENSION="one" WORDSID="w10.137, w10.138" off-
setStart="2271" offsetStop="2285" Color="Chocolate"/>
<RELATION ID="r1" TYPE="referential" SUBTYPE="isa"
TRIGGER="w10.135, w10.136" FROM="e1" TO="e2"/>
```

II. *Spatial relations* with three subtypes are listed and illustrated below:

- *near* (express closeness between entities);
Lângă 1:[Moineşti] se află localitatea 2:[Comăneşti]. / (EN) (Near 1:[Moinesti] is situated 2:[Comanesti].) ⇒ [1] near [2];
- *far* (express distance between entities);
1:[Ţara noastră] … decât de 2:[Oceanul Arctic] / (EN) (1:[Our country] ... than the 2:[Artic Ocean]) ⇒ [1] far [2];
- *position* (position between entities involving cardinal point).
1:[Munții Parâng]… se află la estul 2:[Munților Retezat] / (EN) (1:[Parâng Mountains] … are located east of 2:[Retezat Mountains]) ⇒ [1] position [2];

```
<W Case="direct" Definiteness="yes" Gender="masculine"
ID="w158.5" LEMMA="munte" MSD="Ncmpry" Number="plural"
POS="NOUN" Type="common" offsetStart="21084" off-
setStop="21090" text="Munții"/>
      <W ID="w158.6" LEMMA="parîng" MSD="Np" POS="NOUN"
Type="proper" offsetStart="21091" offsetStop="21097"
text="Parâng"/>
      ...
<W Case="direct" ID="w158.12" LEMMA="sine" MSD="Px3--r"
POS="PRONOUN" Person="third" Type="reflexive" off-
setStart="21148" offsetStop="21150" text="se">
```

```
<W EXTRA="tranzitiv" ID="w158.13" LEMMA="afla" MSD="Vmis3s" Mood="indicative" Number="singular" POS="VERB" Person="third" Tense="past" Type="predicative" offsetStart="2151" offsetStop="21155" text="aflǎ">
<W Case="direct" Definiteness="yes" Gender="masculine" ID="w158.16" LEMMA="munte" MSD="Ncmpry" Number="plural" POS="NOUN" Type="common" offsetStart="21165" offsetStop="21172" text="Munții" />
<W Case="direct" Definiteness="no" EXTRA="ParticipleLemma:reteza(tranzitiv)" Gender="masculine" ID="w158.17" LEMMA="retezat" MSD="Afpmsrn" Number="singular" POS="ADJECTIVE" offsetStart="21173" offsetStop="21180" text="Retezat"/>
<ENTITY ID="e60" TYPE="landform" SUBTYPE="mountain" DIMENSION="many" WORDSID="w158.5, 156.6" offsetStart="21084" offsetStop="21097" color="DimGray"/>
<ENTITY ID="e61" TYPE="landform" SUBTYPE="mountain" DIMENSION="many" WORDSID="w158.16, 156.17" offsetStart="21145" offsetStop="21160" color="DimGray"/>
<RELATION ID="r43" TYPE="spatial" SUBTYPE="position" TRIGGER="w158.12, w158.13" CARDINAL="EST" FROM="e60" TO="e61"/>
```

III. *Structural relations* with four subtypes are listed and illustrated below:

- *vicinity* (neighbors).

1:[România] are o deschidere de 234 km la 2:[Marea Neagră] / (EN) (1:[Romania] has an opening of 234 km to 2:[Black Sea]) ⇒ [1] vicinity [2];

- *part-of* (one entity is part of another).

1:[România] concentrează pe teritoriul său două treimi din lanţul 2:[Munţilor Carpaţi] / (EN) (1:[Romania] concentrates on its territory two thirds of the 2:[Carpathian Mountains]) ⇒ [2] part-of [1];

- *confluent-of* (branch for rivers).

Bazinul 1:[Mureşului]… 2:[Târnava], cel mai important affluent. / (EN) (1:[Mures] Bay ... 2:[Tarnava], most important confluent.”) ⇒ [2] confluent-of [1];

- *source* (root of rivers)

1:[Târnava Mică], cu izvoare în 2:[Munţii Gurghiu] / (EN) ([Târnava Mică], with source in 2: [Gurghiu Mountains].) ⇒ [1] source [2];

```
<W ID="w668.15" LEMMA="târnava" MSD="Np" POS="NOUN" Type="proper" offsetStart="84349" offsetStop="84356" text="Târnava"/>
<W Case="direct" Definiteness="no" Gender="feminine" ID="w668.16" LEMMA="mic" MSD="Afpfsrn" Number="singular" POS="ADJECTIVE" offsetStart="84357" offsetStop="84361" text="Micǎ"/>
```

```
<W ID="w668.17" LEMMA="," MSD="COMMA" POS="" off-
setStart="84361" offsetStop="84362" text=","/>
<W ID="w668.18" LEMMA="cu" MSD="Sp" POS="ADPOSITION" off-
setStart="84363" offsetStop="84365" text="cu"/>
<W Case="direct" Definiteness="no" Gender="feminine"
ID="w668.19" LEMMA="izvor" MSD="Ncfprn" Number="plural"
POS="NOUN" Type="common" offsetStart="84366" off-
setStop="84373" text="izvoare"/>
<W ID="w668.20" LEMMA="în" MSD="Sp" POS="ADPOSITION" off-
setStart="84374" offsetStop="84376" text="în"/>
<W Case="direct" Definiteness="yes" Gender="masculine"
ID="w668.21" LEMMA="munte" MSD="Ncmpry" Number="plural"
POS="NOUN" Type="common" offsetStart="84377" off-
setStop="84383" text="Munții"/>
<W ID="w668.22" LEMMA="gurghiu" MSD="Np" POS="NOUN"
Type="proper" offsetStart="84384" offsetStop="84391"
text="Gurghiu"/>
<ENTITY ID="e135" TYPE="water" SUBTYPE="river"
DIMENSION="one" WORDSID="w668.15, 668.16" off-
setStart="84349" offsetStop="84361" color="DarkGreen"/>
<ENTITY ID="e136" TYPE="landform" SUBTYPE="mountain"
DIMENSION="many" WORDSID="w668.21, 668.22" off-
setStart="84377" offsetStop="84391" color="DimGray"/>
<RELATION ID="r90" TYPE="structural" SUBTYPE="source"
TRIGGER="w668.18, w668.19" FROM="e135" TO="e136"/>.
```

4 Statistics and interpretation

From the entire set of entities presented above, our Corpus highlights the values for the 20 entity types. Because it doesn't have yet an explicit notation entirely accepted by the annotators' group, we will restrict to these partial results (see table 2), for two reasons: first, because we relied on the high precision of accepted types of entity (person, location, organization) and on the high precision of resulted tags from the process with POS-Tagger [20] and the second reason is that we wanted to accelerate the manual annotation process, which certainly resulted in many errors as well (e.g. insignificant difference between *cultural* and *tourism* entities, etc.).

Table 2 compares the automatic detection of the most general types of entities which the annotator easily discovers in the manually annotated corpus. With these results, Precision, Recall and F-Measure, we could calculate the corresponding values, after a new check of annotation for the other types. At this moment, the empirical results for automatic detection of the mentioned types of entity are significant. This shows that a classification of geographical entities not resulting in subjective interpretations will be necessary.

Table 2. Automatic and manual entities annotation results

Type	Precision	Recall	F-measure
location	81.25%	83.25%	82.24%
geo_position	83.50%	85.00%	84.24%
person	82.50%	84.00%	83.24%
organization	87.00%	90.25%	88.60%

5 Conclusions and Future Work

This research is a preliminary study in semi-automatic recognition of geographical entities representing the basis for defining the semantic relationships between them. Classification and annotation of entities has meant a long process which is leading to promising results.

The described analysis is a statistic method which proved efficient, and combines the empirical results obtained from manual and automatic annotation. Our current corpus is now Gold Corpus, this kind of study being intended for the automatic recognition of entities for any kind of text with geographical topic. Moreover, the NER was extended in respect of entity diversity, as well as database in existing gazetteers. So far we identified three types of semantic relations for a geographical text, which had been manually annotated. In the future, we aim at defining patterns containing information at lexical and syntactic level to discover these semantic relations and why not, define other types of semantic relation. Besides the presented semantic annotation, the following remain to be discussed: negative relations (e.g. România…, nu face parte din Peninsula Balcanică / (EN) Romania..., is not part of the Balkan Peninsula) or position in the past of some toponyms (e.g. Generația orașelor antice… Histria, Tomis, Callatis, Apulum, Ampelum, Napoca, Potaissa, Sucidava / (EN) Antic city generation … Histria, Tomis, Callatis, Apulum, Ampelum, Napoca, Potaissa, Sucidava), etc.

Acknowledgement. This work was financed by the Romanian Ministry of Education and Research (UEFISCDI) under the Partnerships Programme (PN II Parteneriate, competition PCCA 2013), project code: PN-II-PT-PCCA-2013-4-1878..

References

1. Bartalesi Lenzi, V., Moretti, G., Sprugnoli, R.: “CAT: the CELCT Annotation Tool”, LREC (2012)
2. Chen, W.-T., Styler, W.: “Anafora: A Web-based General Purpose Annotation Tool”, ACL, paper N13-3004 (2013)
3. Cristea, D., Gîfu, D., Colhon, M., Diac, P., Bibiri, A., Mărănduc, C., and Scutelnicu, A.-L.: “Quo Vadis: A Corpus of Entities and Relations” in Language, Production, Cognition, and the Lexicon. Text, Speech and Language Technology, Part VI - Language Resouces and Langauge Engineering, Nuria Gala, Reinhard

Rapp and Gemma Bel-Enguix (eds.), vol. 48/2015, New York, USA, pp. 505-543 (2015)

4. Cunningharn, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N, Roberts, I., Gorell, G., Funk, A., Roberts, A. et al.: “Developing Language Processing Components with Gate Version 8” (2014)
5. Gîfu, D. and Vasilache, G.: “A language independent named entity recognition system” in M. Colhon, A. Iftene, V. Barbu Mititelu, D. Cristea, and D Tufiş (eds.), "Alexandru Ioan Cuza" University Publishing House, Iaşi, pp. 181-188 (2014)
6. Ignat, E.: “RARE-UAIC (Robust Anaphora Resolution Engine)”, open-resource on META-SHARE, “Alexandru Ioan Cuza” Univrsity of Iași (2011)
7. Kim, J.-D., Ohta, T., Pyysalo, S., Nguyen, N., Bossy, R., Tsujii, J.: “Overview of BioNLP Shared Task 2011”, ACL (2011)
8. Levi, J. N.: The Syntax and Semantics of Complex Nominals. Academic Press, New York (1978)
9. Li, H., Krause, S., Xu, F., Uszkoreit, H., Hummel, R., Mironova, V.: “Annotating Relation Mentions in Tabloid Press”, LREC (2014)
10. Mani, I., Harris, D., Hitzeman, J., Quimby, R., Richer, J., Weller, B.: “SpatialML: Annotation Scheme, Resources and Evaluation”, LREC (2008)
11. Morton, T., and LaCivita, J.: “WordFreak: An Open tool for linguistic Annotation”, ACL (2003)
12. Năstase, V., Nakov, P., Séaghdha, D., Szpakowicz, S.: “Semantic Relations Between Nominals”. Morgan & Claypool Publishers, California (USA) (2013).
13. Neguț, S., Apostol, G., Ielenicz, M.: “Geografie”, Humanitas Educațional, București (2008)
14. Nguyen, N., Tsujii, J., Kim, J.-D.: “Overview of the Protein Coreference task in BioNLP Shared Task 2011”, ACL (2011)
15. Ohta, T., Pyysalo, S., Tsujii, J., Ananidou, S.: “Open-domain Anatomical Entity Mention Detection”, NACTEM (2012)
16. Séaghdha, D.Ó , Copestake, A.: “Semantic classification with distributional kernels”. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08), Manchester, UK (2008)
17. Simionescu, R.: “UAIC Romanian Part of Speech Tagger”, resource on nlp-tools.info.uaic.ro, “Alexandru Ioan Cuza” University of Iași (2011)
18. Simionescu, R.: Romanian deep noun phrase chunking using graphical grammar studio. In: Moruz, M. A., Cristea, D., Tufiş, D., Iftene, A., Teodorescu, H. N. (eds.) Proceedings of the 8th International Conference “Linguistic Resources And Tools For Processing Of The Romanian Language”, 135–143 (2012)
19. South, B.R, Shen, S., Leng, J., B.Forbush, T., DuVall, S.L., and Chapman, W.W.: “A prototype tool set to support machine-assisted annotation”. In Proceedings of the 2012 Workshop on Biomedical Natural Language Processing, BioNLP 12, pp. 130–139, Stroudsburg, PA, USA. Association for Computational Linguistics (2012)
20. Tufiş, D. and Dragomirescu, L.: “Tiered Tagging Revisited” in Proceedings of the 4th LREC Conference, Lisabona, pp. 39-42 (2004)

A Survey on Statistical-based Parallel Corpus Alignment

Sulema Torres-Ramos, Raymundo E. Garay-Quezada

Universidad de Guadalajara, Departamento de Ciencias Computacionales, Jalisco, México
sulema.torres@cucei.udg.mx, raymundo.garay@alumnos.udg.mx

Abstract. The text alignment is an important process of different Machine Translation systems. This task consists in identifying correspondences between words, sentences or paragraphs of a source text and their translation (parallel corpus). There are two main approaches to perform parallel corpus alignment: the statistical-based methods and lexical-based methods. In this paper, the main statistical-based methods for align parallel corpus are presented.

Keywords: Statistical Corpus Alignment, Parallel Corpus, Machine Translation, Natural Language Processing.

1 Introduction

Natural Language Processing (NLP, also known as Computational Linguistics) is a field of Computer Science which aims to create and understand the language that human beings use to communicate. NLP has a lot of important tasks and applications; one of them is Machine Translation, which aims to automatically translate a text from one language to another one.

To make Machine Translation possible, there are several techniques based on dictionaries, statistics or examples. Even though these techniques have their own advantages and disadvantages, they share some methods like text alignment process [3].

Text alignment process consists in organize parallel corpus in order to establish a correspondence between paragraphs, sentences and/or words [16] of the source texts and their translations. Parallel corpus can be defined as two sets of texts in different languages where one of these sets is the source text and the other one is their translation.

There are two main approaches to perform parallel corpus alignment: the statistical-based methods and lexical-based methods. Lexical-based approaches rely on existing lexical knowledge such as antonyms and synonyms, translations of a word, etc.; while Statistical-based approaches rely on non-lexical information, such as sentence length, sentence position, co-occurrence frequency, sentence length ratio in two languages, etc. [13].

In this paper, it will be addressed some algorithms which use statistical information to align parallel corpora; from algorithms which can be considered the first in their field such as Gale and Church, Brown, and K-vec algorithm, their upgrades like Va-

pp. 57–76; rec. 2015-01-20; acc. 2015-03-10

nilla, Moore’s Association-based algorithm, and DK-vec algorithm. And, some recent algorithms like Bleualign and its Iterative version.

2 Machine Translation

Translation is an ancient human activity that consists on communicates a message from an “original” language into a “terminal” language always taking care of not change the idea meaning. At late 1940s, once the digital computers were developed, Warren Weaver had the idea that, the computers could perform automatic translation; he conceives the problem of automatic translation as cryptography problem [30]. Since then, many efforts were made on this new task both in hardware, by improving memory and access store resources, as in software with the dictionary-based translation as its principal approach [29].

By 1966, a publication of the ALPAC reported, as a result of an investigation, that machine translation had no future but, the computational linguistic, and machine-aided translation look promising; the later resulted on a low investments on former task [25].

In despite of ALPAC report, many researchers kept their effort on developing machine translation systems, looking for the best translation arguing that, a post-edition was economically viable [29]. As a result, many system were developed during this period of time, most of them categorized as the Rule-based Machine Translate approach (RBMT), giving rise to the resurgence of machine translation task, during 1980, and beginning with the development of two new approaches, known as Example-based machine translation (EBMT) [22], and statistical machine translation (SMT) [1], the latter based on Weaver's very first ideas on machine translation.

2.1 Machine Translation Paradigms

2.1.1 Ruled-based

The Rule-based machine translation strategy has an explicit linguistic knowledge base, i.e. a linguistic expert builds the necessary grammatical rules to perform a better translation, however, the translation effectiveness vary depending on the deepness of the logical representation of the sentences. The deeper the rule abstraction, the more complex the task of mapping a sentence to its translation [29].

2.1.2 Example-based

This approach is based on tagged corpus used as an example-database. A sentence to be translated is compared with a database of examples to identify the similar sentences. In order to achieve this, a sentence is aligned against several examples-templates, then, the more similar template, based on the alignments, is used to retrieve the possible translations [29].

2.1.3 Statistical

As the Example-based, the Statistical approach use corpora in order to achieve the translation but instead of compute the similarity among sentences, this approach employ two process: training and decoding. As all the supervised algorithms, this one use a set of examples in order to create a statistical model that represents the target language, then, when a translation is requested, the correct translation is search in the space of all the possible translation learned in the statistical model, until find the one with the better probability [29].

3 Text Alignment

To make the text alignment process possible, we need to use a *written corpus*[1] [17]. This can be defined as a collection of texts or a huge text that is used for research, especially for developing translation software and natural language processing [3]. Corpora can be labeled or unlabeled. On one hand, labeled corpora are annotated to identify various attributes or linguistic information such as the topics or themes of the documents contained in the corpora, or the part of speech of the words, among others. For example, the labeled attributes in a corpus for the word "roses" could be *noun*, *plural*, etc. Linguistic information that could be labeled would be its *lemma*[2], the correct sense of the word according to a specific dictionary, etc.; and, in other languages like Spanish, labels like *feminine noun* or *masculine noun* could be added. Figure 1 shows a sample of the SemEval-2015 task 13 corpus [20] which consists in four documents taken of European Medicines Agency documents, the KDE manual corpus and the EU bookshop corpus.

```
<?xml version="1.0" encoding="UTF-8" ?>
<corpus lang="en">
<text id="d001">
<sentence id="d001.s001">
<wf id="d001.s001.t001" pos="X">This</wf>
<wf id="d001.s001.t002" lemma="document" pos="N">document</wf>
<wf id="d001.s001.t003" lemma="be" pos="V">is</wf>
<wf id="d001.s001.t004" pos="X">a</wf>
<wf id="d001.s001.t005" lemma="summary" pos="N">summary</wf>
<wf id="d001.s001.t006" pos="X">of</wf>
<wf id="d001.s001.t007" pos="X">the</wf>
<wf id="d001.s001.t008" lemma="european" pos="J">European</wf>
<wf id="d001.s001.t009" lemma="public" pos="J">Public</wf>
<wf id="d001.s001.t010" lemma="assessment" pos="N">Assessment</wf>
<wf id="d001.s001.t011" lemma="report" pos="N">Report</wf>
<wf id="d001.s001.t012" pos="X">(</wf>
<wf id="d001.s001.t013" lemma="epar" pos="N">EPAR</wf>
<wf id="d001.s001.t014" pos="X">)</wf>
<wf id="d001.s001.t015" pos="X">.</wf>
</sentence>
```

Figure 1. A sample of the corpus used for the task 13 in SemEval-2015.

[1] From now on, written corpus will be addressed as corpus or corpora

[2] Is the canonical form, dictionary form, or citation form of a word

This corpus has been tagged with a special notation that identifies /sentence/word/lemma/pos/id. On the other hand, unlabeled corpora have no linguistic information and do not have a defined structure; it is often user-generated information such as email or instant messages, documents or social media postings.

There are different types of corpus (see Figure 2). In the field of machine translation, the classification of *monolingual* and *multilingual corpus* is important:

1. Monolingual corpora are texts in only one language.
2. Multilingual corpora are texts in multiple languages, and they can be divided in the following sub-categories:
   a. *Parallel corpus* can be defined as two sets of texts in different languages where one of these sets is the source texts and the other one is their translations. Each of these set of texts can be known as *bitexts* [11]. Parallel corpora can be uni-directional, bi-directional, or multi-directional [15]. For example, the Bible and its copies in different languages can be considered parallel corpus.
   b. On the other hand, a *comparable corpus* is a set of texts in different languages that share the same main topic but they differ in the way they address it. It means that a comparable corpus is not a source text and their translation. According to Simões Branão [28] "a set of new articles, from journals or news broadcast systems, as they refer the same event in different languages can be considered Comparable Corpora".

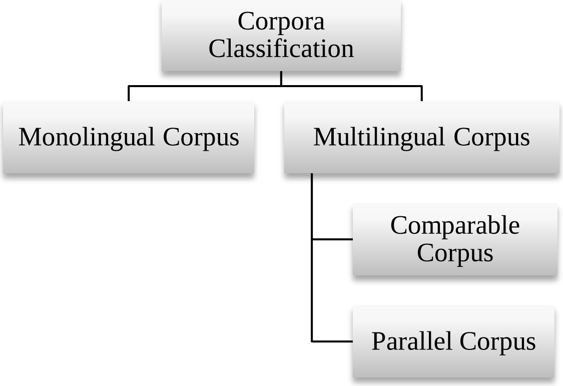


Figure 2. Corpora Classification

Corpus alignment consists in organize parallel corpus in order to establish a correspondence between paragraphs, sentences and/or words [16] of the source texts and their translations. Yet the automatic alignment of parallel corpora is not a trivial task for some language pairs [24].

E. Macklovitch [16], recognized 4 levels of alignments (see Figure 3):

- 1st level alignment: this level addresses the alignment of the whole text when the text is not long enough.

- 2nd level alignment: this level recounts the alignment of paragraphs.
- 3rd level alignment: this level describes the alignment of sentences.
- 4th level alignment: this level relates the alignment of words between bitexts.

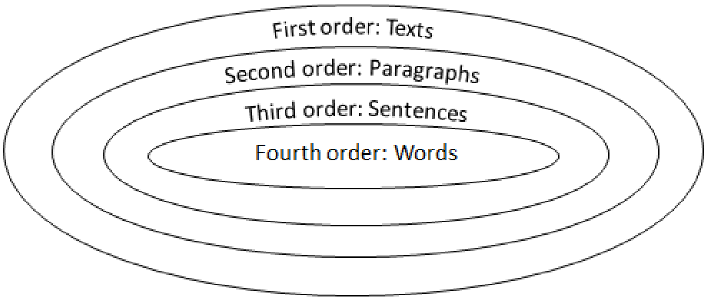


Figure 3. Alignment levels.

3.1 Corpus Alignment Approaches

There are two main approaches to perform parallel corpus alignment: one approach is based on statistical data, while the other one applies additional linguistic knowledge. The basis of this distinction is related with the kind of data being processed independently of the methods of processing [10]. Several techniques have been developed based on these approaches, each with their own advantages and disadvantages.

Lexical-based approaches rely on existing lexical resources, such as large-scale bilingual dictionaries and glossaries [13], to obtain information about the languages, such as antonyms and synonyms, translations of a word, etc. Some of these techniques are presented in [10][4][12][14][18][13]. These techniques tend to be slower than the techniques based on statistic information and are dependent language. The main disadvantage of these techniques is that their performance depends heavily on the lexical information used on the alignment process. However, many of these methods are being developed because are expected to generate better results than the statistical ones [6].

Statistical-based approaches, which are the basis of this study, rely on non-lexical information, such as sentence length, sentence position, co-occurrence frequency, sentence length ratio in two languages, etc. [13]. These techniques make the alignment process faster, and are, generally, independent language. However, the main disadvantage of these techniques is that their performance depends heavily on the structural similarity between target text and source text of the bitexts. According to [13] “the attraction of these resource-poor approaches arises from the sharp contrast between their poor resources and their rich outcomes”.

Figure 4 shows the statistical-based methods discussed in this survey and some lexical-based methods as a reference.

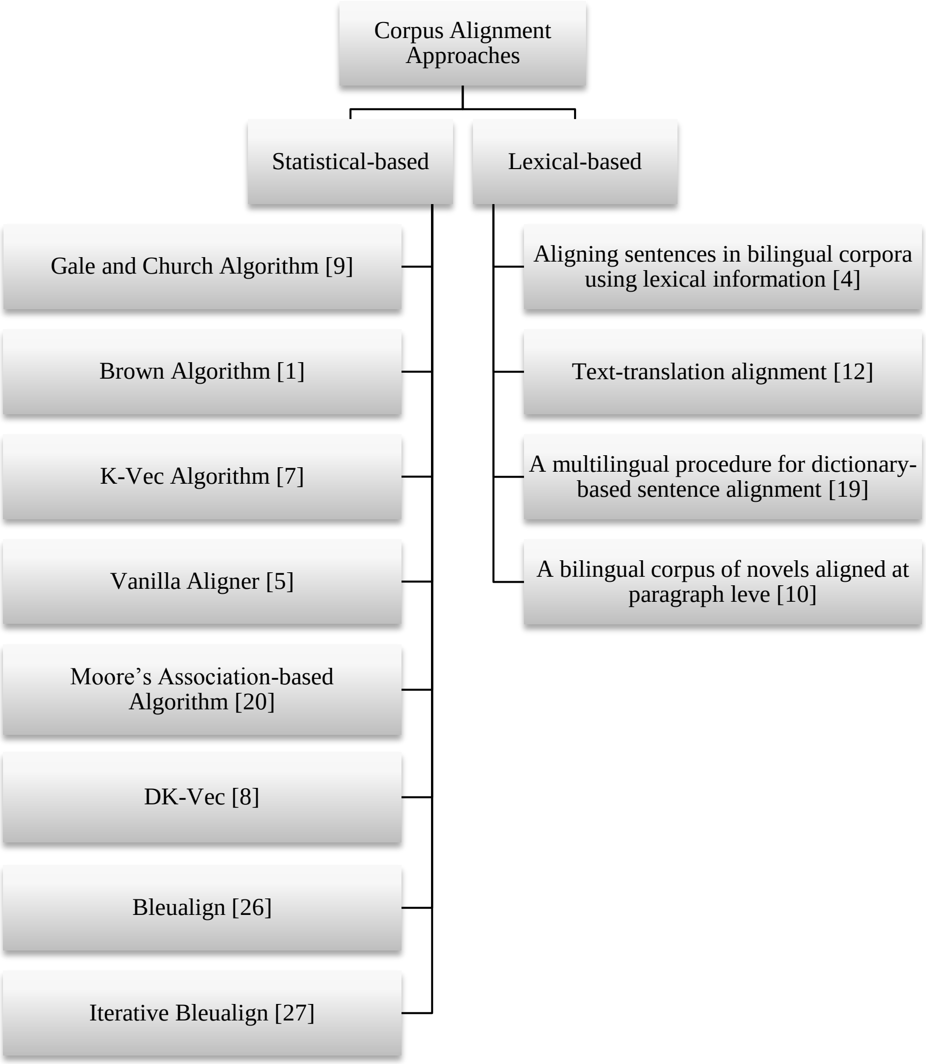

Corpus Alignment Approaches
Statistical-based
Lexical-based
Gale and Church Algorithm [9]
Brown Algorithm [1]
K-Vec Algorithm [7]
Vanilla Aligner [5]
Moore's Association-based Algorithm [20]
DK-Vec [8]
Bleualign [26]
Iterative Bleualign [27]
Aligning sentences in bilingual corpora using lexical information [4]
Text-translation alignment [12]
A multilingual procedure for dictionary-based sentence alignment [19]
A bilingual corpus of novels aligned at paragraph leve [10]

Figure 4. Some of the Statistical- and Lexical-based methods for parallel corpus alignment

4 Statistical-based Methods for Parallel Corpus Alignment

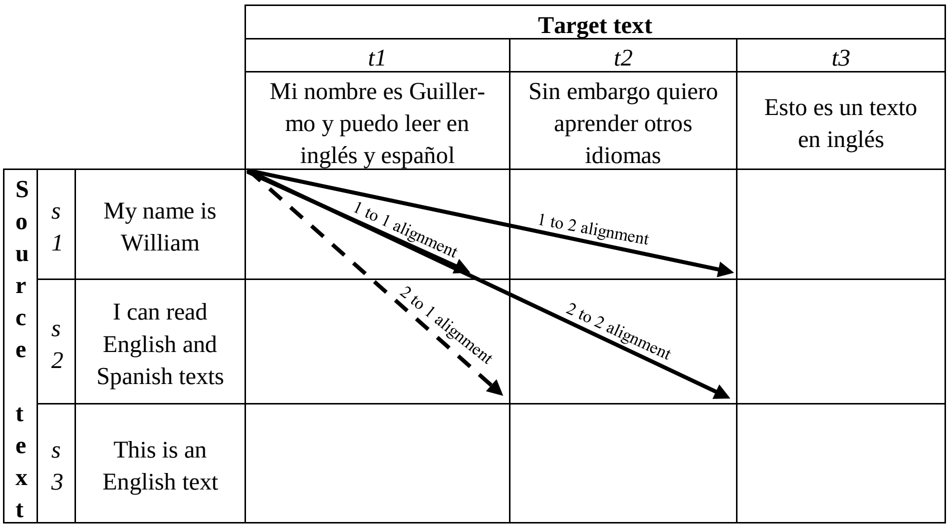
4.1 Gale and Church Algorithm

The main idea behind this sentence level aligner is “longer sentences in one language tend to be translated into longer sentences in the other language, and shorter sentences tend to be translated into shorter sentences” [9]. A parallel corpus already aligned in paragraphs is required by this algorithm.

This algorithm considers the length of the sentences (in characters) to align them. These are used to calculate a value called *distance measure* for each pair of sentences (one of the source texts and one of the target texts). The lower the distance measure, the higher the probability that the sentences correspond themselves. The algorithm considers the following alignment categories [3]:

1. 1-to-1 alignment: this is the best possible scenario, where one sentence of a language (source text) is translated into exactly one sentence in the other language (target text).
2. 1-to-2 alignment: one sentence in the source text is divided into two sentences in its translation.
3. 1-to-0 alignment: one sentence in the source text is not translated in the target text.
4. 0-to-1 alignment: one sentence in the target text was added by the translator.
5. 2-to-2 alignment: two sentences in the source text correspond with two sentences in its translation.
6. 2-to-1 alignment: two sentences in the source text correspond with only one sentence in its translation. This is because the two sentences were merged by the translator.

Each alignment is considered to find the correspondence of each sentence. For instance, as Figure 5 shows, the length of the sentence *s1* (15 characters) is not similar than the length of *t1* (45 characters) nor the length of the combination *t1* y *t2* (84 characters). However, the length of the combination *s1* y *s1* (47 characters) and the length of *t1* (45 characters) are more alike. Therefore, the best option, is this case, is the 2-to-1 alignment.



			Target text		
			t1	*t2*	*t3*
			Mi nombre es Guillermo y puedo leer en inglés y español	Sin embargo quiero aprender otros idiomas	Esto es un texto en inglés
Source text	*s 1*	My name is William			
	s 2	I can read English and Spanish texts			
	s 3	This is an English text			

Figure 5. Alignment example using Gale and Church Algorithm

Even though the number of characters in a sentence is used as distance measure in the example, the algorithm uses the following distance measure (D) [9]:

$$D = -100 * \log(2 * \left(1 - Prob(\delta)\right)) \tag{1}$$

Where $Prob(\delta)$ is defined as:

$$Prob(\delta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\delta} e^{\frac{z^2}{2}} \, dz \tag{2}$$

And δ is defined as:

$$\delta = \frac{l_2 - l_1 c}{\sqrt{l_1 s^2}} \tag{3}$$

Where:

- l_2 and l_1 are the lengths of the portions of text
- c also known as *mean,* is the expected number of characters in l_2 per character in l_1. This value is equal to 1 to simplify the algorithm.
- s^2 is the variance of the number of characters in l_2 per character in l_1. This value is equal to 6.8.

The values of mean and variance are estimated to European languages (English, French, German, Spanish, and etcetera). They could change in other pairs of languages that are not alike, for example, Spanish and Japanese.

4.2 Brown Algorithm

This sentence level aligner was developed by Brown *et. al.* [1]. Similar to Gale and Church, this algorithm considers the length of the sentences to align them, but the difference is that the measure sentence length is in words. They use the tags included in the TEX format of the Canadian Hansard corpus as anchor points to help in the alignment process. The algorithm considers major and minor anchor points and performs the alignment in two steps:

1. Aligning the major anchors:
   (a) To each possible section alignment, a cost is assigned, rewarding the exact matchings and penalizing omissions and inexact matchings.
   (b) Determine the alignment with the least total cost, using dynamic programming. The output is an aligned sequence of sections.
   (c) Sections that not contains the same number of minor anchors in the same order for each corpus are eliminated
2. Aligning the minor anchors:
   (a) Divide the remaining sections (not accepted in the previous step) into subsections.

(b) The algorithm aligns the sentences using a Hidden Markov Model based on the number of words.

4.3 K-vec Algorithm

This algorithm is a word level aligner developed by Pascale Fung and Kenneth Ward Church [7]. K-vec has a very important characteristic; it does not depend on sentence boundaries (for example, periods "." are boundaries in European languages). Using this feature, the algorithm aims to align languages that are not alike; for example, English and Japanese, as well as European languages. The main idea of this algorithm is "if two words are translations of each other, they are more likely to occur in the same segments than two words which are not" [8].

In order to determine if a word is the translation of another one in a bitexts, this algorithm focuses on the similarity of their distributions in the corresponding text. First, the algorithm splits the parallel texts into k number of segments. For each word in the text, a K-dimensional binary vector is created and it indicates the presence (value 1) or absence (value 0) of that word in each segment. Note that the first position of the vector corresponds to the first segment of the text, the second position to the second segment, and so on; also, the frecuency or position of the word in each segment is not important. For example, Figure 6 shows parallel texts (Spanish and English) divided into 3 segments (k=3). In this case, the Spanish text has the words "casa" and "computadora", while English text has "house" and "computer". The word "casa" appears in segments 1 and 3 of the Spanish corpus, so the corresponding vector would be **V***spanish-casa* = <1, 0, 1>. The word "computadora" appears in segment 2 of the Spanish corpus, so the corresponding vector would be **V***spanish-computadora* = <0, 1, 0>. The word "house" appears in segments 1 and 3 of the English corpus, so the corresponding vector would be **V***english-house* = <1, 0, 1>. The word "computer" appears in segment 2 of the English corpus, so the corresponding vector would be **V***english-computer* = <0, 1, 0>.

The resulting vectors represent the distribution of each word in their corresponding corpus.

Once the distributions of the words have been calculated, the algorithm computes the similarity for each pair of vectors (one of the source text and one of the target text). The higher the similarity, the higher the probability that the words correspond themselves.

In the previous example the vectors **V***spanish-casa* and **V***english-house* are identical so it can be considering that the word "house" is the English translation of the Spanish word "casa", and vice-verse. On the other hand, the vectors corresponding to "casa" and "computer" are different, therefore it can be determined that "computer" is not the translation of "casa". Nevertheless, calculating the similarity of pairs of vectors goes far beyond simply identifying whether they are equal or not. In the original algorithm Pointwise Mutual Information and T-Score are used as measures of association in order to compute the similarity between pair of vectors (words) [7].

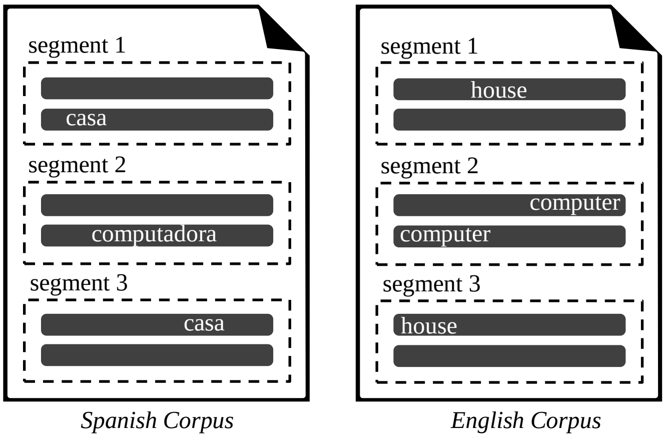


Figure 6. Example of the K-vec algorithm.

4.4 Vanilla Aligner

Vanilla aligner was presented by Pernilla Danielson and Daniel Ridings in 1997 [5], and is an upgrade of Gale and Church algorithm [9]. The same way that its predecessor this is a sentence level aligner and dependent on sentences boundaries. The main contribution of this aligner is the compatibility to work with bitexts in SGML format. One of the benefits to using bitexts in SGML format is that a standard form or structure can be settled. Therefore it will help to identify sentence boundaries more easily.

Even though Vanilla aligner is based on the Gale and Church aligner using the sentences length (character count) to find correspondences between them, this algorithm implements some modifications to the alignment process:

1. Parsing the bitexts in SGML format: Searches for all spaces and replaces them with an end of line character.
2. Reading through a normalized SGML file and creates input files for the aligner:
   (a) Concatenates all the lines of a paragraph within a <BODY> element to one single line
   (b) Looks for sentence ending punctuation marks and stick an “end of sentence” after each one
   (c) Removes the SGML labels and the double spaces resulting
   (d) Adds the “end of paragraph” code.
3. Finally two files are created. One per language. The words are separated by line and contain labels to point out the sentence and paragraph ending.

The two files generated are the input for the aligner. The Gale and Church aligner used the same input data. Vanilla Aligner also uses the same sentence alignment categories: 1-to-1, 1-to-2, 1-to-0, 0-to-1, 2-to-2, and 2-to-1.

Besides the pre-processing to work with SGML format, another difference between Vanilla aligner and Gale and Church algorithm is the output file, post-processing and the access to the results. First, Vanilla aligner generates only one output file while Gale and Church algorithm creates two output files. According to Danielsson and Ridings [5] "having two separate files to work with instead of only one, makes it slightly more inconvenient to check the results and look for possible errors". Second, they address some ways of dealing with results. For example, the use of some labeled text indicating the correspondence between sentences, or the use of databases like mSQL.

4.5 Association-based Bilingual Word Alignment

There are two main motivations in this work: first, they strongly believed that word-based alignment could be a good startup for phrase-based alignment; second, the algorithms presented until then, for example, Brown *et. al.*[1] presents the disadvantage of high computational complexity, and that it was able to find low computational complexity strategies, but with a proportional good accuracy.

Moore [20] presents three different strategies for word alignment: 1) 1-to-1 word type alignment, 2) n-to-1 alignment, and 3) Token alignment selection. Each strategy has two or more methods to overcome several problems. And all of them are based on Log-Likelihood-Ratio (LLR) association measure which has been used in constructing lexicon translation.

1. 1-to-1 word alignment: The following methods only permit one-to-one alignment and do not take position into account.

   (a) Method 1

   It used the Competitive linking algorithm [18] and use the LLR score as a measure. The algorithm consists in three steps:

   i. Find the word type pair that have the highest association score (LLR) of any pair of words type have not been linked.
   ii. Add one to the count of linked occurrences of this pair of word types, and subtracts one to the unlinked count instances.
   iii. Repeat while words can be linked

   (b) Method 2

   The problem with the later method is that the sentence alignment decision, given a pair of words, is taken independently for the same pair of word in different sentences. This method considers this, and to solve this problem, a second alignment is applied using the conditional probability of a pair of words as an alignment score. The new alignment is defined as follows:

   i. Count the number of links in the training corpus for each pair of words linked in any sentence pair.
   ii. Count the number of co-occurrences in the training corpus for each pair of words linked in any sentence pair by Method 1
   iii. Compute the link probability score [20] for each pair of words linked in any sentence pair by Method 1.

iv. Align sentences pairs by competitive linking using link probability score.

(c) Method 3

The method 2 fails to display monotonic relationships between precision and recall. So, a discounted link probability (LPd) [20] is applied. The algorithm is the same that the one in method 2 but using LPd.

2. n-to-1 alignment: This technique is based on the method 3 of the One-to-one alignment. It is shown that iterative application of such a method could create Many-to-One clusters by building of clusters, incrementally.

3. Token alignment selection methods: This method is a complement to the 1-to-1 and n-to-1 alignment. This addresses the problem of selecting the best word token alignment for a given word type alignment, *i.e.*, the incorporation of positional information into associated-based word alignment.

   (a) Method A

   Make a random choice (without replacement) for each word type, in the alignment, from among tokens of that type.

   (b) Method B

   A word token alignment consisted with a given word type alignment that is the most monotonic is found, by minimizing the degree of no monotonicity of such alignment. If there is more than one word token with the less degree of no monotonicity, then it is picked arbitrarily.

4.6 DK-vec Algorithm

The Dynamic K-vec algorithm, DK-vec for now on, is based on its ancestor, the K-vec algorithm, which works under the assumption that two words are more likely to appear in the same segment if they are translations of each other. However, it usually does not happen with languages that are not alike. In addition, k-vec algorithm does not consider *a priori* information from the language or corpus characteristics reducing its performance [8].

To overcome those disadvantages, the authors propose the DK-vec algorithm, which includes two important characteristics. First, they define the concept of arrival interval, which is the difference between the initial position of a word in a segment and the next appearance of the same word. The set of arrival intervals is known as *recency information*. Second, a pattern matching technique is proposed to gives the algorithm the capability of align vectors of different lengths.

The algorithm works as follow, first, it is necessary to define the position of a word, defined as the number of characters counted from the beginning of the document until the first character of the intended word. For example, if we have the text “This is an example” the position of the word “*an*” is the ninth given that there are nine characters between the first character in the text, which is “t”, and the first character of the word "*an*" which is "a", see Figure 7. It is worth knowing that, when the length of a text is computed, each blank character is counted as one. Then, the *recency information* is calculated using the position vector computed before.

Text:	***T***	***h***	***i***	***s***		***i***	***s***		***a***	***n***		***E***	***x***	***a***	***m***	***p***	***l***	***e***
No. Character:	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Word Position:	✓					✓			✓			✓						

Figure 7. Position of the words (DK-vec algorithm).

Now, suppose that we want to compute the arrival vector for the word "*example*" of a given corpus. First, it is necessary to build the position vector of the word which is [12, 100, 250, 500, 700, 800], i.e., the first character of the word "*example*" appears in the positions 12 of the corpus, the second in position 100 and so on. Then, the recency information vector is built based on the position vector computed before obtaining the following vector [88, 150, 250, 200, 100]. The length of the first vector is equal to the number of times that the word appears in the whole corpus, and the length of the second vector is the length of the first vector minus 1, see Figure 8.

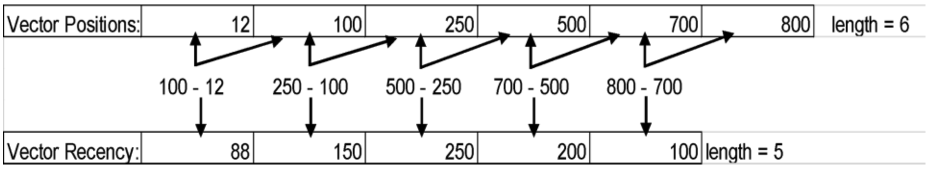


Figure 8. Vector positions and Vector recency of the word "example"

To analyze the information it is possible to represent the two vectors, position vector and arrival vector, as a signal creating a graph where the vertical axe is the recency information and the horizontal axe is the word position information.

Now suppose there are three pairs of vectors corresponding to three different words, being two of them its corresponding translation. Their vectors and graphical representation are seen as follows.

1. Word "example"
   (a) $\mathrm{V}_{\mathrm{Positions}} =< 12,100,250,500,700,800 >$
   (b) $\mathrm{V}_{\mathrm{Recency}} =< 88,150,250,200,100 >$
2. Word "ejemplo"
   (a) $\mathrm{V}_{\mathrm{Positions}} =< 50,140,280,520,710,805,860 >$
   (b) $\mathrm{V}_{\mathrm{Recency}} =< 90,140,240,190,95,55 >$
3. Word "ver"
   (a) $\mathrm{V}_{\mathrm{Positions}} =< 10,100,540,900,1500 >$
   (b) $\mathrm{V}_{\mathrm{Recency}} =< 90,440,360,600 >$

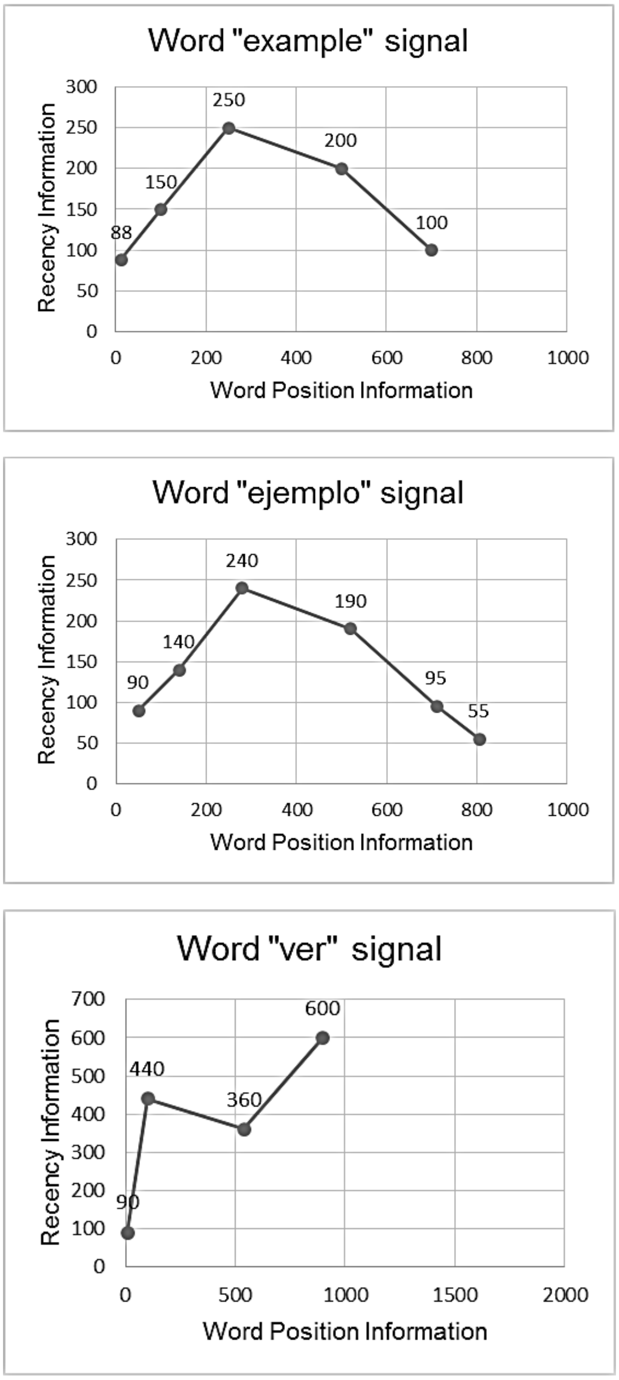


Figure 9. Graphs of the words "example", "ejemplo" and "ver"

In the graphical representation (Figure 9), it is visually clear that two of them are more alike, the ones corresponding to the words “example” and “ejemplo”, while they are very different from the graph representing the “ver” word. In order to make a quantitative representation, the Dynamic Time Warping algorithm is used [8].

4.7 Bleualign

This sentence aligner was created by Rico Sennrich and Martin Volk [26]. Its main idea is to use a Machine Translation (MT) system and a translation evaluator (BLEU, hence its name) to help in the alignment process.

In order to better understand the aligner, it is necessary to know about BLEU. BLEU is an algorithm which evaluates the quality of an MT output. For the purpose of measure that quality, BLEU obtains a score by comparing the MT output against one or more reference human translations. “The closer a machine translation is to a professional human translation, the better it is” [23]. The BLEU score goes from *0* to *1*. A score *1* means that the MT output is identical to its reference.

This alignment algorithm receives the following input data: source text and target text. The input data have delimiters to divide the text. The algorithm performs the following steps [26]:

1. Translate the source text to the corresponding target language using an MT system.
2. Based on the BLEU algorithm, each sentence of the translated source text is compared to each sentence of the target text to determine the similarity score between pair of sentences.
3. The algorithm keeps the 3 best-scoring alignment candidates for each source sentence.
4. Choose the combination of 1-to-1 alignments that maximizes the BLEU score, using dynamic programming. The result is an ordered list of 1-to-1 alignments.

In the last steps 1-to-1 alignments are made, therefore there may be sentences in both texts (source and target) that remained unaligned. In order to align those sentences, the algorithm does the following:

5. Determine if the 1-to-1 alignments are n-to-1 alignments: For each pair of aligned sentences (*i*, *j*), the sentence of the translated source text (*i*) is concatenated with its nearest unaligned sentences neighbors (according to the ordered list obtained in step 4, the all possible 1-, 2- or 3- sentence sequences). Then, those sentence sequences are compared with the target sentence (*j*). If one of those comparisons obtains a best BLEU score against the (*i*, *j*) score, the algorithm chooses a new n-to-1 alignment (instead of 1-to-1 alignment). Else, the process is performed, analogous, with the sentence of the target source text (*j*) against the sentence sequences (concatenations with sentences neighbors) of the translated source text. The latter determines the 1-to-n alignments.

6. As final resource, the algorithm tries to align the remaining sentences using the Gale and Church algorithm [9]. The input to this algorithm is the automatic translation of the source text and the target text because "this gives slightly better results, and should be more robust for unrelated language pairs, for which a length-based comparison is less suited" [26].

4.8 Iterative Bleualign

This sentence-aligner algorithm was created in 2011 by Rico Sennrich and Martin Volk [27] as a result of a deeper analysis of the disadvantage on using machine translating system in the process of alignments. Sennrich and Volk established that MT-based alignments strongly depends on the correct translation of the source text, and given that MT systems are generally fed with aligned texts, then it is evident the existence of a circular dependency.

In order to overcome this dependency, this algorithm presents a bootstrapping approach to do the alignment. The sentence alignment consists on the following steps:

1. Align parallel text.
   (a) First iteration: An implementation of any sentence alignment tool that does not require additional source is used. This work uses the Gale and Church algorithm [9].
   (b) Subsequent iteration:
      (i) The translation of the source text is made using the alignment of the previous step and a SMT system,
      (ii) Then, with the later translation, Bleualign is used for the alignment.
2. Train the SMT system on the sentences-aligned corpus

In SMT, it is common that the alignment algorithms produce several misaligned sentence pairs, however, it is not a big problem given that wrong phrases translation tends to be less probable than the corrects ones. Nevertheless, the latter is not true for an iterative approach where the training test is also the to-be-translated text. In order to overcome this problem, a pruning strategy was implemented. It consist on computing whether the occurrence frequency of phrase pairs in the SMT is statistically significant, or to be expected by chance

5 Discussion

As it was mentioned before, the algorithms presented through this work use statistical information to perform corpus alignment. During the next section, the main idea, the input and output parameters, and the assumptions of each algorithm, will be analyzed.

The Gale and Church algorithm and Brown algorithm were ones of the first corpus aligners that used statistical information. The simplistic idea was used both, as a platform for posterior algorithms or as an alternative to others. To assess their performance, Gale and Church algorithm uses a trilingual corpus of the Union Bank of Switzerland of economic results. Gale and Church reported the following results using a character sentence-length: “there was a 4.2% error rate on 1316 alignments, averaged over both English-French and English-German data”. In addition, an alternative test was performed using a word sentence-length; however, the results were not good enough showing a 6.5% error rate against the 4.2% of the ones of character sentence-length. In regard to the initial condition and parameters, the algorithm requires some probabilistic values that were assigned based on Canadian Hansard bitexts but these values didn’t change a lot on others kind of corpus, so they remain as constants. Moreover, the input data used in this algorithm requires a very specific structure, so it needs some pre-processing step on the parallel corpus.

On the other hand, Brown uses the Canadian Hansard corpus to evaluate its algorithm. Similar to Gale and Church, Brown algorithm have achieved remarkably good outputs for language pairs like English-French with error rates of 4% on an average. However, these algorithms are not robust with respect to non-literal translations and deletions; and they depend heavily on the delimiters (paragraphs or anchor points). Also, the algorithms present bad performance with languages that are not alike.

The Vanilla aligner is very similar to the Gale and Church aligner; in fact, it is used as a base with some adjustments. The main differences were the compatibility with a labeled or formatted text and the way the output data is presented. The former difference refers to the pre-processing step in which labels are removed, allowing to the algorithm have more parallel corpus to work with. The second refers to the way the algorithm presents the output information. The Gale and Church aligner creates two files with the alignment results while in Vanilla aligner one file is generated. According to [5] “this algorithm gets it right more than 95% of the time. When it does go wrong, it is usually when it tries to find a 0-1 alignment (or a 1-0) that should be 3-1 or 1-3, for example.”

The basic idea of Gale and Church aligner works when the parallel corpus to be aligned are European languages, for example; English, Spanish French, among others. Nevertheless, it appears to be false when we want to align languages like English against Chinese, i.e., from different language familiy. Several important characteristics have been identified, for example, their alphabet, which makes the sentence-length be different; the sentence boundaries, like the linguistic boundaries and so on. All these reasons may lead to other researchers to create new aligner algorithms, for example, Moore’s Association-based algorithm, K-vec and DK-vec algorithm.

Moore made two main contributions, the first was the development of faster algorithm, and second, he used word-association statistics for sentence alignment. In addition, Moore claims that, even though he cannot ensure that the word-association heuristics are better than the well-funded alignment approaches, this work give an insight on that word-association heuristics is still a good research opportunity.

The K-vec algorithm is a word level aligner that is not dependent to linguistic boundaries. It aims to align texts with different language ancestor, for instance, Japanese and Spanish. In fact, this aligner creates its own boundaries in the bitexts called segments. These K segments are the information vectors representing each word in the text, i.e., one vector per word. The corpus used to assess the performance of the algorithm was Canadian Hansards in order to have a previous comparison reference.

Finally, as the authors say, one of the principal contributions of this work is that "could be used as a starting point for more detailed alignment algorithm…" given that it generates “a quick-and-dirty estimate of a bilingual lexicon”[7].

Shortly afterwards, a new algorithm was developed, called DK-vec, which is based on the K-vec algorithm and its principal contribution is the addition of statistical information, as a new vector known as arrival vector, in order to improve the parallel corpus alignment. Unlike the K-vec algorithm, DK-vec divides the bitexts into K pieces and for each word, two vectors are computed, the vector position and the arrival vector, with a dimension k and k-1, respectively. These vectors are treated as signals and used to measure the signal similarity using dynamic time warping. As a result, it presents a better performance on corpus alignment of different root languages like Japanese against English corpora.

On 2010, the Bleualign does two main contributions: First, a new approach, known as MT-based sentence alignment, is created and consists of the use of an MT system to improve the sentence alignment. Second, Bleualign first uses the BLEU score as a similarity measure for sentence alignment. However, this new approach has a high dependency on the MT-System used for the translation. Problems such as dependency rise when the MT-System cannot translate the pair of languages resulting in worst performance (from 50% to 61% of accuracy) than Gale and Church algorithm (from 68% to 80% of accuracy) over the Text+Berg corpus. However, if it is used a reliable MT-System, the performance of this aligner is about 81% to 95% of accuracy in the same bitexts. Also, it has been shown that if we try to align a translation made by MT-System and the target text in Gale and Church aligner, the performance gets better from 68% - 80% to 72% - 83% of accuracy [26].

One year later, the same authors of the Bleualing algorithm created a new version of the algorithm. In this case, the authors point out that Bleualing is not 100% reliable with a pair of languages that does not have a good MT-System and they tried to solve it with this new algorithm. In this case, they use the same three elements of the previous algorithm; MT-System, BLEU-score and Gale and Church aligner. However, they change the order of these elements to solve this dependency problem. The authors did the experiments over the same bitexts (Text+Berg corpus). This new aligner had a performance of 69.5% to 94.4% accurate [27] and overcome the problem of use an MT system for sentence alignment problem, which is a basic tool of almost every MT system.

6 Conclusion

In this paper, some of the most representative statistical-based algorithms to parallel corpus alignment were described and a discussion about their results, advantages and disadvantages was presented.

There is not one statistical-based approach that works for all kinds of languages in the scope of parallel corpus alignment, i.e., some methods perform better when the languages to be alignment share a common ancestor, but others are more robust under this condition.

Even though these approaches have achieved remarkably good results, considering the poor resources used, there is much to improve. More recent works suggest using the lexical and statistical information to improve the performance of the parallel corpus alignment.

Acknowledgments. Work done under partial support of Mexican Government (CONACYT, SNI) and PRODEP (project 227835).

References

1. Brown, P. F., Lai, J. C., & Mercer, R. L. (1991, June). Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics* (pp. 169-176). Association for Computational Linguistics.
2. Brown, P. F., Cocke, J., Della Pietra, S., Della Pietra, V. J., Jelinek, F., Mercer, R. L., & Roossin, P. S. (1988, June). A Statistical Approach to French/English Translation. In *RIAO* (pp. 810-829).
3. Cendejas Castro, E.A. (2013). *Alineación automática de textos paralelos a nivel de palabras información lingüística diversa* (Ph.D. thesis). Centro de Investigación en Computación-IPN, México.
4. Chen, S. F. (1993, June). Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics* (pp. 9-16). Association for Computational Linguistics.
5. Danielsson, P., & Ridings, D. (1997, February). Practical presentation of a “vanilla” aligner. In *TELRI Workshop in alignment and exploitation of texts,* February.
6. Félix, J. Á. V., & Sidorov, G. (2004). Proyecto de preparación del corpus paralelo alineado español-inglés. *Memorias del 5o encuentro internacional de computación ENC-2004*, Colima, México, 235-242.
7. Fung, P., & Church, K. W. (1994, August). K-vec: A new approach for aligning parallel texts. In *Proceedings of the 15th conference on Computational linguistics*-Volume 2 (pp. 1096-1102). Association for Computational Linguistics.
8. Fung, P., & McKeown, K. (1994). Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. *arXiv preprint cmp-lg/9409011*.
9. Gale, W. A., & Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics*, *19*(1), 75-102.
10. Gelbukh, A., Sidorov, G., & Vera-Félix, J. Á. (2006). A bilingual corpus of novels aligned at paragraph level. In *Advances in Natural Language Processing* (pp. 16-23). Springer Berlin Heidelberg.
11. Harris, B. (1988). Bi-text, a new concept in translation theory. *Language Monthly*, 54, 8-10.
12. Kay, M., & Röscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, 19(1), 121-142.
13. Kit, C., Webster, J. J., Sin, K. K., Pan, H., & Li, H. (2004). Clause alignment for Hong Kong legal texts: A lexical-based approach. *International Journal of Corpus Linguistics*, *9*(1), 29-51.
14. Langlais, P., Simard, M., & Véronis, J. (1998, August). Methods and practical issues in evaluating alignment techniques. In *Proceedings of the 36th Annual Meeting of the Asso-*

ciation for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 711-717). Association for Computational Linguistics

15. McEnery, A., & Xiao, R. Z. (2008). Paralell and comparable corpora: what are they up to?. *Incorporating Corpora: Translation and the Linguist. Translating Europe. Clevendon: Multilingual Matters*.
16. Macklovitch, E., & Hannan, M. L. (1998). Line ‘em up: advances in alignment technology and their impact on translation support tools. *Machine Translation*, 13(1), 41-57.
17. Marín, F. M. (1993). La Biblioteca Electrónica en el Archivo Digital de Manuscritos y Textos Españoles. *Lexis: Revista de lingüística y literatura*, *17*(1), 33-56.
18. Melamed, I. D. (2000). Models of translational equivalence among words. *Computational Linguistics*, *26*(2), 221-249.
19. Meyers, A., Kosaka, M., & Grishman, R. (1998, October). A multilingual procedure for dictionary-based sentence alignment. In *Conference of the Association for Machine Translation in the Americas* (pp. 187-198). Springer Berlin Heidelberg.
20. Moore, R. C. (2005, June). Association-based bilingual word alignment. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts* (pp. 1-8). Association for Computational Linguistics.
21. Moro, A., & Navigli, R. (2015). SemEval-2015 task 13: multilingual all-words sense disambiguation and entity linking. *Proc. of SemEval-2015*.
22. Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. *Artificial and human intelligence*, 351-354.
23. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311-318). Association for Computational Linguistics.
24. Piao, S. (2000). *Sentence and Word Alignment between Chinese and English* (Ph.D. thesis). Lancaster University, Lancaster.
25. Pierce, J. R., & Carroll, J. B. (1966). *Language and machines: Computers in translation and linguistics*.
26. Sennrich, R., & Volk, M. (2010, November). MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, Denver, Colorado.
27. Sennrich, R., & Volk, M. (2011, May). Iterative, MT-based sentence alignment of parallel texts. In *18th Nordic Conference of Computational Linguistics, NODALIDA*
28. Simões, A. (2004). *Parallel corpora word alignment and applications* (master thesis). Universidade do Minho, Braga.
29. Way, A., & Hearne, M. (2011). On the Role of Translations in State-of-the-Art Statistical Machine Translation. *Language and Linguistics Compass*, 5(5), 227-248.
30. Weaver, W. (1955). Foreword: the new tower. In W.N. Locke and A.D. Booth (Editors), *Machine Translation of Languages: Fourteen Essays*, Cambridge, MA.

Arabic Probabilistic Context Free Grammar Induction from a Treebank

Nabil Khoufi, Chafik Aloulou and Lamia Hadrich Belguith

ANLP Research Group, MIR@CL Lab,
Faculty of Economics and Management,
University of Sfax, Tunisia.
{nabil.khoufi, chafik.aloulou, l.belguith}@fsegs.rnu.tn

Abstract. Linguistic resources are very important to any natural language processing task. Unfortunately, the manual construction of these resources is laborious and time-consuming. The use of annotated corpora as a knowledge database might be a solution to a fast construction of a grammar for a given language. In this paper, we present our method to automatically induce a syntactic grammar from an Arabic annotated corpus (The Penn Arabic TreeBank), a probabilistic context free grammar in our case. To construct our resource, we first induce context free rules from the annotated corpus trees as a first step and then we calculate a specific probability for each induced rule. Finally, we present and discuss the obtained grammar.

Keywords: Linguistic Resource Construction, Syntactic Grammar, PCFG, Arabic language, PATB corpus.

1 Introduction

The development of effective NLP applications requires the use of reliable and large linguistic resources (knowledge) such as lexicons or grammars. For example, the parsing task requires, in addition to the input sentence, some knowledge about the kind of syntactic analysis that should be produced as output. One method to provide such knowledge to the parser is to write down manually a grammar of the language.

However, manual construction of such linguistic resources is a difficult task to undertake, and it is time consuming. Unlike a programming language, natural language is far too complex to simply list all the syntactic rules. Moreover, it is difficult to exhaustively list lexical properties of words in addition to validate the written grammar by some linguists of the treated language.

In this regard, describing the Arabic language needs special attention given its greater ambiguity when compared to other natural languages (especially the Indo-Europeans languages). Indeed, Arabic has several characteristics among which we mention the following:

pp. 77–86; rec. 2015-01-25; acc. 2015-02-27

- Vocalic ambiguity: the oversight of the vocalization marks increases the ambiguity of words' comprehension;
- Grammatical ambiguity: several words may have the same grammatical interpretation;
- Agglutination: articles, prepositions, pronouns, etc. can be affixed to adjectives, nouns, verbs and particles to which they are related;
- Problems related to the segmentation of texts into sentences;
- Abundant use of recursive structures in the Arabic texts;
- Elliptic and anaphoric structures.

An alternative method to build linguistic resources is to use treebanks as source of knowledge. Indeed, treebanks, as rich corpora with annotations, provide an easy way to build other linguistic resources, such as extensional and intentional lexicons, syntactic grammars, bilingual dictionaries, etc. This promotes their reuse as well as makes explicit their implicit information. Furthermore, treebanks have many advantages: they are not only developed and validated by linguists, but also submitted to consensus, which promotes their reliability. Having such resource makes it possible to generate automatically and in a very controlled basis, new and wide coverage resources on other formalisms. These resources inherit the original treebank qualities, while reducing construction time. Since we have access to the PATB corpus, currently, the largest Arabic corpus in modern standard Arabic, we have decided to use it in order to build our syntactic grammar.

This paper is organized as follows: Section 2 is devoted to presenting related works on the construction of grammars for Arabic language. Section 3 gives the probabilistic context free grammar's (PCFG) basic definitions. Section 4 explains our method. Experimental results are presented and discussed in section 5. Section 6 provides the conclusion and future work.

2 Grammars for Arabic Language

Grammar studies the way in which words/morphemes join to form meaningful sentences. It consists of a set of constraints on the possible sequences of symbols expressed as rules or principles. The construction of grammars is an important task to several NLP applications. Thus, for the Arabic language, many grammars were built following various grammar formalisms. The main purpose of a grammar in NLP is to serve the parsing task.

In this section, we present the different grammars that have been designed to represent Arabic syntax. All these works attempt to cover most of the syntactic rules of the Arabic language and its rich linguistic phenomena.

Habash and Rambow [1] have built a tree adjoining grammar by extracting elementary trees from an annotated corpus. In order to accomplish this, the authors reinterpreted the PATB as a dependency corpus and then extracted the Tree-Adjoining Grammar from the corpus. They used part 1, version 2.0 of the PATB, which comprises around 160,000 words of annotated Arabic text from newswire sources. Extracted structures have some variations in their constituent's positions because of the semi-fixed order of the Arabic language. Thus, some obtained sentences have VSO (Verb Subject Object) structure and others have SVO structure.

The LFG (Lexical Functional Grammar) [2] formalism has been used to represent Arabic language syntax. Attia [3] used this formalism to build the grammar. Indeed, the developed grammar focuses on the specific features of Arabic, especially nominal sentences (sentences without verbs), sentences with a hidden subject, verbal sentences etc. Both LFG structures (c-structures and f-structures) are built to each sentence described by this grammar.

The HPSG (Head driven Phrase Structure Grammar) [4] was also used to describe the syntax of Arabic in the work of Haddar [5]. The main objective of this work is to construct an Arabic HPSG grammar based on a proposed type hierarchy that categorizes Arabic words. In fact, some adaptations were introduced to HPSG at the level of features and ID schemata. All linguistic resources (e.g., lexicon, type hierarchy, syntactic rules) are specified in the Type Description Language (TDL). The experimentation of the constructed grammar was achieved using the Linguistic Knowledge Building (LKB) platform which contains generation tools. The authors justified the use of the IDL by its syntax similarity to HPSG representation.

In the present work, we chose to induce a PCFG from a treebank. The grammar choice is justified by its capacity to give a partial solution for any grammar ambiguity; indeed a PCFG gives some idea of the plausibility of a sentence. Likewise, a PCFG is a robust grammar that can admit everything with low probability. In the following section, we present the basic definitions of the PCFG grammar and then we present our method.

3 PCFG Basic Definitions

A probabilistic context-free grammar (PCFG) also called stochastic CFG (SCFG), is an extension of the famous context-free grammar, where a certain probability is assigned to each rule. Probabilistic context-free grammars are defined by a 5-uplet $< N, T, R, S, P>$ as follows:

- N is a finite set of non-terminal symbols.
- T is a finite set of terminal symbols.
- R is a finite set of rules Ri of the form $X \rightarrow Y1Y2 \dots Yn$, where $X \in N$, $n \geq 0$, and $Yi \in (N \cup T)$ for $i = 1 \dots n$.
- $S \in N$ is a distinguished start symbol.
- P is the set of probabilities Pi associated with rules Ri where: $\sum P(X \rightarrow Yi) =1, \forall X \in N$ *and* $Yi \in (N \cup T)$ *for* $i = 1 \dots n$.

Note that some sentences may have more than one underlying derivation in case of the use of a classic CFG and therefore generates several parse-trees. So the probabilities P in a PCFG are used to determine which parse-tree is the most likely to be the best parse for the given sentence. The probability of a parse-tree is obtained by multiplying the probability of each used rule at each node of the tree.

4 Proposed Method

Our objective is to automatically induce a PCFG from an annotated corpus. This mechanism consists of two steps: The first one is to induce a CFG rules from the annotated corpus. The second step is to allocate a specific probability for each induced rule. The application of these two steps allows us to obtain a PCFG. Figure 1 illustrates the workflow of our method for deriving a PCFG.

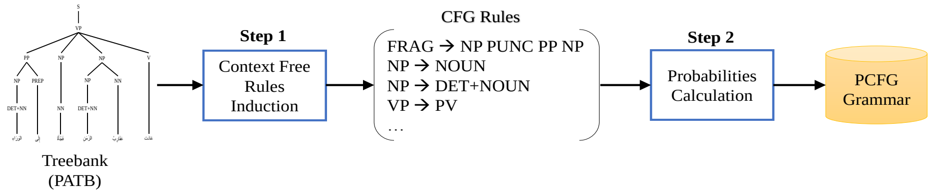


Fig. 1. Steps of the PCFG induction method

Given that our method is based on an annotated corpus to build the PCFG grammar, we present in the next subsection the used corpus then we detail the steps of the method.

4.1 The Used Corpus

In our work, we chose to use the well-known corpus, Penn Arabic Treebank (PATB). This choice was motivated not only by the richness, the reliability and professionalism with which it was developed but also by the syntactic relevance of its source documents (converted to several other Treebank representations). Indeed, its annotations have the advantage of being reliable. This is proven by its efficiency in a large number of works in various fields of NLP [6]. The good quality of the text and its annotations are proven by their pertinence for the creation of other Arabic Treebanks such as the PADT [7] and the CATiB [8], which converted the PATB to its syntactic representations in addition to other annotated texts.

Indeed, these annotations were manually elaborated and validated by linguists. Moreover, this treebank is composed of data from linguistic sources written in Modern Standard Arabic. This corpus is also the largest Arabic corpus which integrates syntactic tree files. The use of a large amount of annotated data in a construction process of a grammar increases the quality of the generated linguistic resource.

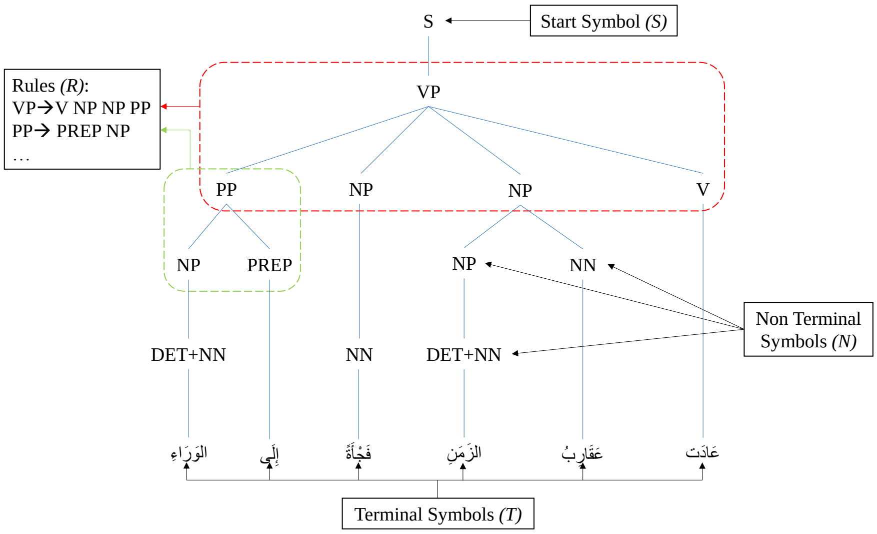
The PATB was developed in the Linguistic Data Consortium (LDC) at the University of Pennsylvania [9]. Texts in the corpus do not contain any vowels as it is typically in use in most texts written in Arabic.

4.2 Description of the Method

As shown in Figure 1, the first step of the method is the induction of CFG rules, including duplicates which will be used in the second step. A deep study of the PATB allows us to identify rules which guide the CFG rules induction process. Indeed, we focused on the morpho-syntactic trees of the PATB and we identified the following rules:

R1: Tree root → Start symbol
R2: Internal tree node → Non terminal symbol
R3: Tree word → Terminal symbol
R4: Tree fragment → CFG rules

In fact, we noticed that each parse tree is a sequence of context-free rules and each one has the same symbol “S” at its root. Thus, the root symbol “S” is taken to be the start symbol (S) of the grammar. Non-terminal (N) symbols consist of the set of internal nodes of the whole parse tree. The set of all words seen in the trees (leaves) represent the terminal symbols (T). Edges between the nodes of the trees are used to induce the CFG rules(R). The following figure presents the process of induction of the PCFG elements (S, N, T, R) from a PATB parse tree.



(The hands of time turned suddenly backwards.)

Fig. 2. Induction of PCFG elements from a PATB parse tree

Note that the tags on the right hand-side of the induced rules are in reversed order compared to the parse tree. This is due to the reading orientation of the Arabic language which is from right to left.

In the Arabic language, the word and its determinant are agglutinated as we can see in the word الزَمَنِ (Noun). We chose to keep these elements together in one rule to reduce the grammar size. This choice does not influence the analysis quality since there is no loss of information.

Noun → Det Noun
Det → ال
Noun→زمن

→

Noun → Det+Noun
Det+Noun → الزمن

After applying the first step of our method to the example of Figure 2, we obtain 12 CFG rules composed by 6 contextual rules and 6 lexical rules as presented in the following table:

Table 1.CFG Rules induced from the example of Figure2

Contextual Rules	**Lexical Rules**
S→VP	V→عَادَتْ (turned)
VP→V NP NP PP	NN→عَقَارِبُ (the hands)
NP→NN NP	DET+NN→الزَّمَنِ (time)
NP→DET+NN	NN→فَجْأَةً (suddenly)
NP→NN	PREP→إِلَى (to)
PP→PREP NP	DET+NN→الوَرَاءِ (backwards)

Once the CFG rules had been induced from the PATB with all duplicates, we move to the second step, which consists in the rule probability calculation *(P)* to finally obtain the PCFG grammar. Each rule probability is estimated using the following formula:

$$P(X \to Y) = \frac{Count(X \to Y)}{Count\ (X)}$$

Where Count (X→Y) is the number of times the rule X→Y is seen in the Treebank and Count (X) is the count of rules that have the non-terminal X on the left-hand side.

For example, the rule VP→V NP PP is seen109 times in the PATB and we have counted 1311 rules that have VP on the left-hand side, thus:

$$P(VP \to V\ NP\ PP) = \frac{109}{1311}$$

In the following section, we describe our experiment and expose some interesting information about the obtained grammar.

5 Experimentation and Results

After applying our method, we obtained the PCFG grammar. We used the PATB 3 version 3.2 of this corpus, which consists of 599 files, and includes POS tags, morpho-syntactic structures at many levels and glosses. It comprises 402,291 tokens and 12,624 sentences. It is available in various formats: The “sgm” format refers to source documents. The “pos” format gives information about each token as fields before and after clitic separation. The “xml” format contains the “tree token” annotation after clitic separation. The “penntree” format generates a Penn Treebanking style. And finally the “integrated” format brings together information about the source tokens, the tree tokens, and the mapping between them and the tree structure.

As mentioned earlier, the PATB is a very rich corpus and it contains many annotations such as mood, gender, number, etc. The PATB corpus is annotated using a large set of annotations which gives a high level of granularity. For example, the Part of speech annotation tag set contains 498 tags that provide extensive information: gender, the mood, etc. [9]. There are also 22 syntactic category tags and 20 tags that describe semantic relations between tokens. In addition to that, stop words, which are very numerous in the Arabic language, are also annotated with specific tags.

The incorporation of all this information within the grammar increases its complexity and its size. The size depends on the granularity level of the categories it describes: the higher this level, the more these grammars are complex, but the more they respect the language specificity. For instance, this tag set {ADJ+NSUFF_FEM_DU_NOM, ADJ+CASE_DEF_ACC ADJ+CASE_DEF_GEN ADJ+CASE_DEF_NOM} describes adjectives with a high level of granularity. If the granularity level is reduced to the minimum, these tags will be reduced into one tag, ADJ. This reduction influences the number of grammar rules. We chose to reduce POS tags to the basic tags which are about 70 tags for the Arabic language to facilitate the use of the grammars for NLP application.

There is also other tags in the PATB that are generated during the initial tagging and parsing process like *ICH*, *O*, *RNR*, NONE and NAC. Those tags, if considered in a parsing task, will increase the number of rejected parses because they describe morphologic tagging errors and empty categories. Therefore, those tags were removed from our tag set to maintain a better consistency of our grammar.

We present below some statistics about our PCFG grammar. Table2 presents the most frequent PCFG syntactic rules generated from the PATB corpus after applying our method. Table 3 presents the overall count of rules (contextual and lexical rules).

Table 2. Most frequent LHS rules

Left-Hand symbol (LHS)	NP	VP	S	FRAG	ADJP	UCP	PP
Rule count	1821	1311	1154	360	330	196	150

Table 3. Rule count extracted from the PATB

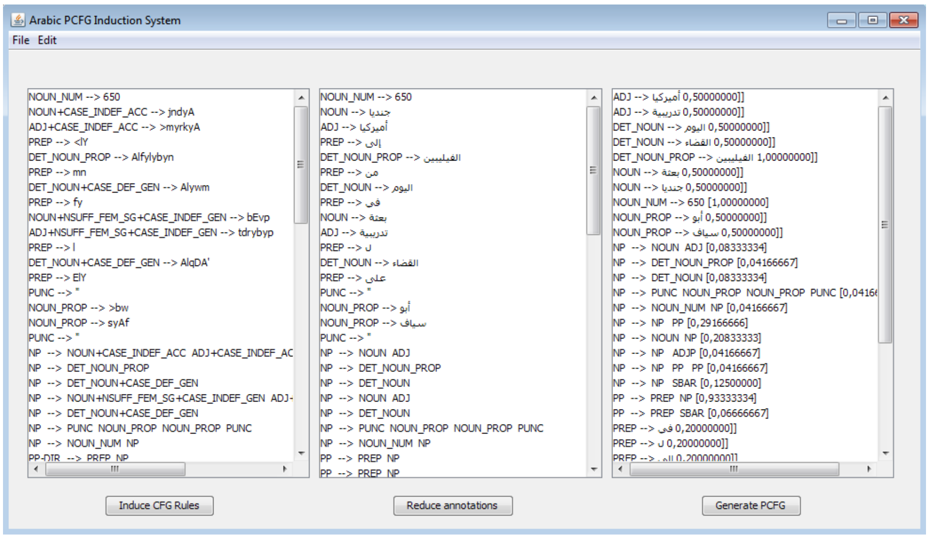
Contextual Rules	5757
Lexical Rules	38 901
Total	44 658

5.1 Arabic PCFG Induction System

We developed a tool to automatically induce the PCFG from PATB tree files. The system allows the user to:

- Induce CFG rules with full morphological annotations,
- reduce the morphological annotations of the induced rules,
- and generate PCFG from the CFG rules

The following figure 3 shows the induction interface of our system.



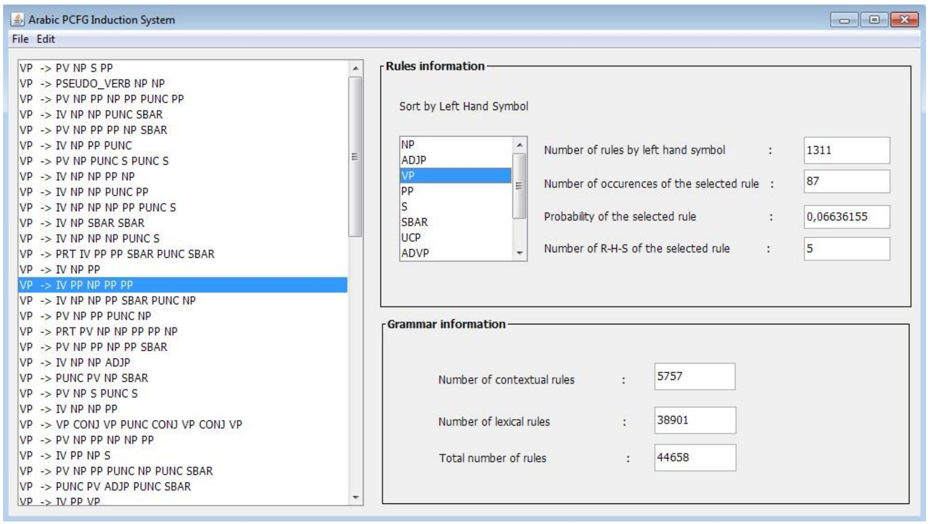
```
NOUN_NUM --> 650
NOUN+CASE_INDEF_ACC --> jndyA
ADJ+CASE_INDEF_ACC --> >myrkyA
PREP --> <lY
DET_NOUN_PROP --> Alfylybyn
PREP --> mn
DET_NOUN+CASE_DEF_GEN --> Alywm
PREP --> fy
NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN --> bEvp
ADJ+NSUFF_FEM_SG+CASE_INDEF_GEN --> tdrybyp
PREP --> l
DET_NOUN+CASE_DEF_GEN --> AlqDA'
PREP --> ElY
PUNC --> "
NOUN_PROP --> >bw
NOUN_PROP --> syAf
PUNC --> "
NP  -->  NOUN+CASE_INDEF_ACC  ADJ+CASE_INDEF_AC
NP  -->  DET_NOUN_PROP
NP  -->  DET_NOUN+CASE_DEF_GEN
NP  -->  NOUN+NSUFF_FEM_SG+CASE_INDEF_GEN  ADJ+
NP  -->  DET_NOUN+CASE_DEF_GEN
NP  -->  PUNC  NOUN_PROP  NOUN_PROP  PUNC
NP  -->  NOUN_NUM  NP
PP-DIR  -->  PREP  NP
```

```
NOUN_NUM --> 650
NOUN --> جنديا
ADJ --> أميركيا
PREP --> إلى
DET_NOUN_PROP --> الفيليبين
PREP --> من
DET_NOUN --> اليوم
PREP --> في
NOUN --> بعثة
ADJ --> تدريبية
PREP --> ل
DET_NOUN --> القضاء
PREP --> على
PUNC --> "
NOUN_PROP --> أبو
NOUN_PROP --> سياف
PUNC --> "
NP  -->  NOUN  ADJ
NP  -->  DET_NOUN_PROP
NP  -->  DET_NOUN
NP  -->  NOUN  ADJ
NP  -->  DET_NOUN
NP  -->  PUNC  NOUN_PROP  NOUN_PROP  PUNC
NP  -->  NOUN_NUM  NP
PP  -->  PREP  NP
PP  -->  PREP  NP
```

```
ADJ --> أميركيا 0,50000000]]
ADJ --> تدريبية 0,50000000]]
DET_NOUN --> اليوم 0,50000000]]
DET_NOUN --> القضاء 0,50000000]]
DET_NOUN_PROP --> الفيليبين 1,00000000]]
NOUN --> بعثة 0,50000000]]
NOUN --> جنديا 0,50000000]]
NOUN_NUM --> 650 [1,00000000]
NOUN_PROP --> أبو 0,50000000]]
NOUN_PROP --> سياف 0,50000000]]
NP  -->  NOUN  ADJ [0,08333334]
NP  -->  DET_NOUN_PROP [0,04166667]
NP  -->  DET_NOUN [0,08333334]
NP  -->  PUNC  NOUN_PROP  NOUN_PROP  PUNC [0,0416
NP  -->  NOUN_NUM  NP [0,04166667]
NP  -->  NP   PP [0,29166666]
NP  -->  NOUN  NP [0,20833333]
NP  -->  NP   ADJP [0,04166667]
NP  -->  NP   PP   PP [0,04166667]
NP  -->  NP   SBAR [0,12500000]
PP  -->  PREP  NP [0,93333334]
PP  -->  PREP  SBAR [0,06666667]
PREP --> في 0,20000000]]
PREP --> ل 0,20000000]]
PREP --> إلى 0,20000000]]
```

Fig. 3. Induction interface

Besides the PCFG induction, the system allows the user to browse the grammar and give some information about it. Indeed, the system can sort grammar rules by the left hand symbol to facilitate the consultation of rules. The system also shows useful information for each selected rule such as the number of rules with the same left hand symbol, the number of occurrences of the selected rule, its probability and the number of right hand symbols in addition to some statistics of the induced grammar.



```
VP  -> PV NP S PP
VP  -> PSEUDO_VERB NP NP
VP  -> PV NP PP NP PP PUNC PP
VP  -> IV NP NP PUNC SBAR
VP  -> PV NP PP PP NP SBAR
VP  -> IV NP PP PUNC
VP  -> PV NP PUNC S PUNC S
VP  -> IV NP NP PP NP
VP  -> IV NP NP PUNC PP
VP  -> IV NP NP NP PP PUNC S
VP  -> IV NP SBAR SBAR
VP  -> IV NP NP NP PUNC S
VP  -> PRT IV PP PP SBAR PUNC SBAR
VP  -> IV NP PP
VP  -> IV PP NP PP PP
VP  -> IV NP NP PP SBAR PUNC NP
VP  -> PV NP PP PUNC NP
VP  -> PRT PV NP NP PP PP NP
VP  -> PV NP PP NP PP SBAR
VP  -> IV NP NP ADJP
VP  -> PUNC PV NP SBAR
VP  -> PV NP S PUNC S
VP  -> IV NP NP PP
VP  -> VP CONJ VP PUNC CONJ VP CONJ VP
VP  -> PV NP PP NP NP PP
VP  -> IV PP NP S
VP  -> PV NP PP PUNC NP PUNC SBAR
VP  -> PUNC PV ADJP PUNC SBAR
VP  -> IV PP VP
```

Rules information

Sort by Left Hand Symbol

NP
ADJP
VP
PP
S
SBAR
UCP
ADVP

Number of rules by left hand symbol : 1311

Number of occurences of the selected rule : 87

Probability of the selected rule : 0,06636155

Number of R-H-S of the selected rule : 5

Grammar information

Number of contextual rules : 5757

Number of lexical rules : 38901

Total number of rules : 44658

Fig. 4. Arabic PCFG browsing interface

6 Conclusion and Future Work

In this paper, we presented our work in inducing an Arabic PCFG from the PATB corpus. As a result, we obtained a new resource which provides a wide coverage and inherits PATB qualities such as its reliability, adherence to consensus and rich annotation. The proposed method consists of two steps: in the first one we induce CFG rules from the PATB parse trees using induction rules. Tag set of the PATB was factorized and filtered to reduce the complexity of the grammar and keeping only useful information. Second we calculated the probability of each rule using the frequency of its occurrence in the corpus. The obtained grammar is composed of 5757 contextual rules and 38 901 lexical rules.

Our main perspective is to test the performance of this grammar in the parsing task. This is a work in progress as we intend to test the obtained grammar using several PCFG parsing algorithms such as Viterbi or CYK. Such comparison between parsing algorithms results on our PCFG is very interesting. This work is part of a hybrid method to parse the Arabic language. The aim of this hybrid method (symbolic / statistical) is the collaboration of two parsers, the first based on a statistical model obtained using supervised learning techniques [10;11] and the second based on the induced grammar described in this paper.

Acknowledgments

We are grateful to Dr. Hela Azaiez, assistant researcher scientist at university of Iowa for proofreading the manuscript.

Reference

1. Habash, N., Rambow, O.: Extracting a tree adjoining grammar from the Penn Arabic Treebank. In: Proceedings of Traitement Automatique du Langage Naturel (TALN-04), pp. 277-284 (2004)
2. Kaplan,R. M., BresnanJ.:Lexical functional grammar: A formal system for grammatical representation. In: J. BRESNAN, Ed., The Mental Representation of Grammatical Relations. Cambridge, Mass, MIT Press, pp. 173-281 (1982)
3. Attia, M. A.: Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation. Doctoral thesis, Faculty of humanities, University of Manchester(2008)
4. Pollard, C., Sag I.: Head-driven Phrase Structure Grammar. CLSI series, University of Chicago(1994)
5. Haddar, K., Boukedi, S., Zalila, I.: Construction of an HPSG grammar for the Arabic language and its specification in TDL. International Journal on Information and Communication Technologies, vol. 3, n° 3, pp. 52-64 (2010)
6. Habash, N. Y.: Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies, pp. 1-187(2010)
7. Hajic, J., Vidová-Hladká, B., Pajas, P.: The Prague dependency Tree-bank: Annotation structure and support. In: Proceedings of the IRCS Workshop on Linguistic Databases, pp. 105-114 (2001)
8. Habash, N., Roth, R. M.: Catib: The Columbia Arabic Treebank. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 221-224, Association for Computational Linguistics (2009)
9. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In: Proceedings of NEMLAR conference on Arabic language resources and tools, pp. 102-109 (2004)
10. Khoufi N., Louati S., Aloulou C., Hadrich Belguith L.: Supervised learning model for parsing Arabic language. In: Proceedings of the 10th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2013), Marseille, France, pp129-136. (2013)
11. Khoufi N., Aloulou C., Hadrich Belguith L.: Chunking Arabic Texts Using Conditional Random Fields. In: Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2014), November, 2014, Doha, Qatar, pp 428-432 (2014)

Towards Logical Inference for Arabic Question-Answering

Wided Bakari,[1] Patrice Bellot,[2] Omar Trigui,[1] Mahmoud Neji[1]

[1] Faculty of Economics and Management, 3018, Sfax Tunisia
[2] Aix-Marseille University, F-13397, Marseille Cedex 20

{wided.bakkari, omar.trigui, Mahmoud.Neji}@fsegs.rnu.tn
patrice.bellot@gmail.com

Abstract. This article constitutes an opening to think of the modeling and the analysis of Arabic texts within a question-answering system. It is a question of exceeding the traditional investigations focused on morpho-syntactic approaches. We present a new approach that analyzes a text, transforms it to logical predicates and extracts the accurate answer. In addition, we represent different levels of information within a text and choose an answer among several proposed. To do so, we transform the question and the text into logical forms. Then, recognize all entailments between them. So, the results of this recognizing are a set of text sentences that can implicate the user's question. Now, our work is concentrated on an implementation step to develop a question-answering system in Arabic using the techniques of textual entailment recognition. Text features extraction (keywords, named entities, relationships that link them) is actually considered the first step in our text modeling process. The second one is the use of textual entailment techniques that relies on inference and logic representation to extract the candidate answer. The last step is the extraction and selection of this answer.

Keywords: NLP, Arabic language, question-answering, recognition text entailment and logic forms.

1 Introduction

Today, we usually weigh multiple questions and need accurate information to answer this question. With the large volume of information found on the internet and the increasing of user's demands, retrieving accurate information is considered a tedious task although the researches in this area do not cease to increase. So, the question-answer process has become one of the most important researches to deal with trustworthy information. It is considered as the heart of the concerns in information retrieval (Sitbon et al., 2006). Generally, the common architecture of a question-answering system (Q-AS[1]) developed is practically a process of three phases (see Figure 1) (Athenikos and Han, 2010), including (question process, document process and answer process).

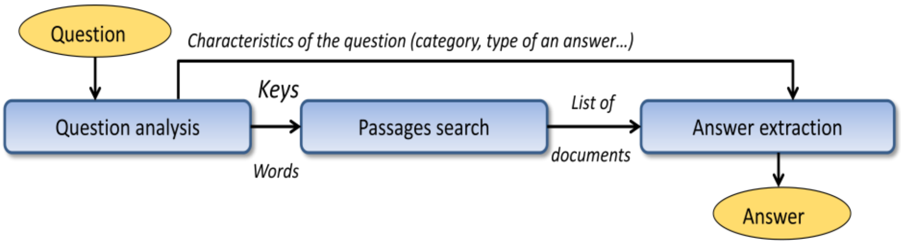


Fig. 1. Architecture of a Q-AS

The potential of question-answering (Wren, 2011) is implemented with the latest success of IBM Watson2 on Jeopardy3. Indeed, Watson analyzes the questions to get what is asked. In less than three seconds, it launched 200 million pages of natural language that contain his memory to find the correct answer and provides evidence for the accuracy of the answer. Apart from the television games, the question-answering technology provided such performance medicine, meteorology, travel, etc.

This article is structured on four sections: Section 2 introduces past works in question-answering systems. Section 3 details our proposed approach for the Arabic and its different steps. Finally, section 4 presents conclusions and potential future directions.

2 Related Works in Question-Answering Systems

Question-answering is a multidisciplinary field where a question-answering system often incorporates techniques and eventual resources, including Information Retrieval IR, Natural Language Processing (NLP), Information Extraction (IE), Machine Learning (ML), etc (Lee et al., 2005). The researches in this area were beginning since 1960. There are many approaches and methodologies that are proposed to many World languages (Bulgarian, Dutch, English, Finnish, French, German, Indonesian, Italian, Japanese, Portuguese, Arabic and Spanish). The question-answering was often occured in conferences and workshops, such as, TREC, CLEF, NTCIR, MUC, etc.

2.1 Question-Answering in English Language

In Natural Language Processing and Information Retrieval, English is the most studied in terms of resources, corpus and systems (Ligozat, 2013). The first question-answering systems focus on English questions came out in 2000. These systems deal with a deep analysis of questions on a restricted or open domain.

[1] Question-Answering System

[2] Watson: http://www-03.ibm.com/innovation/us/watson/.

[3] http://fr.wikipedia.org/wiki/Jeopardy!

QALC (Question Answering program of the Language and Cognition group) (Ferret et al., 2000), (De Chalendar et al., 2002), was developed in the LIMSI laboratory in TREC 1999. This system enables to extract among several candidate sentences the 10 first that provide a suitable and accurate answer. This case is suitable if the selected sentences have different weights. Else, the accurate answer is founded in one of the last sentences as well as in the first ones. Recently, QALC presents an adaptation of an existing question answering system for a machine reading task. Grau and his colleagues proposed an approach for selecting correct answers relies on textual entailment recognition between hypotheses and texts (Grau et al., 2012).

Voorhees provided a question-answering system for factoid questions (Voorhees, 2004). It is the best system in TREC 2004 with a precision of 77% (e.g How many calories are there in a Big Mac?) and a score of 62.2% to answer the question list (e.g list the names of chewing gum).

Somewhere else, AskHERMES[4] (Ask Help clinicians to Extract and aRticulate Multimedia information for answering clinical quEstionS) (Cao et al., 2011) is an online system allows the doctors to ask complex medical questions in consultation and quickly identify the specific answers. This system analyses large volumes of documents to produce short texts as answers. The authors evaluated their system via 4654 medical questions collected in practice. AskHERMES achieved the following two scores 76.0% and 58.0% to classify general themes and extract terms.

Qakis [5] (Question Answering wiKiframework-based System) is a question-answering system in an open field. It generates SPARQL queries from questions, submits them to DBpedia[6] and compares the question to the pattern base to identify the relationship of DBpedia to identify the answer. Also, it provides an integration of multilingual chapters to a DBpedia query with natural language queries. Moreover, QAKiS allows the users to submit a query to a RDF triple store in English and get the answer in the same language. Its architecture is composed of four components (query generation, named entity recognition, pattern-matching and SPARQL package) (Cabrio et al., 2012).

Recently, Tahri and Tibermacine proposed SELNI system which is based on the DBpedia Infobox ontology (Tahri and Tibermacine, 2013). It is focused on the SVM algorithm to train the factoid questions and used SPARQL queries to interrogate the DBpedia server in order to extract the accurate answer. SELNI affected a precision of 86% based on TREC 10 test set.

Finally, the system proposed by (Bhaskar et al., 2012) participates in the main task of QA4MRE@CLEF 2012 and QA4MRE@CLEF 2013. The authors combine the question and each answer option to form the Hypothesis (H) and removed stop words from each hypothesis. This task carried out the associated document based on the TF-IDF of the matching query words along with n-gram overlap of the sentence with hypothesis H. Each sentence allows defining the Text T and the pair (T, H) allocates a ranking score based on textual entailment; each sentence is assigned an inference score with respect to each answer pattern. Based on those ranking a weighing validation is automatically assigned to each answer. The chosen answer is

[4] http://www.askhermes.org/

[5] http://qakis.org/qakis/

[6] http://dbpedia.org

the one that receives the highest score among the list of option answers. This system has a precision of 0.53 and c@1 of 0.65.

2.2 Question-Answering in Chinese and Japanese Languages

Chinese is the second most popular language in question-answering; it was performed for the first time in 2005 at NTCIR (Lee et al., 2005).

Marsha, question-answering system, relies on the same techniques used in English systems developed from TREC (Li and Croft, 2001). Marsha focused on a method based on the TREC question-answering. It has the same performance as some English question-answering systems in the TREC 8 track. Li and Croft have used 51 queries from which 51 they selected 26 queries out of 240 questions collected from Chinese students. The rest of queries are specified to reformulate a question or ask other slightly different. Marsha contains specific techniques dealing with the Chinese characteristics (word segmentation, ordinals processing). It is composed of three modules, such as, query processing, Hanquery search engine and answer extraction.

Some researchers are interested in both English and Chinese questions. They investigate an English-Chinese and Chinese-English translation using Machine Translation systems. This investigations participate in the CLQA (Cross Language Question Answering) task at NTCIR (Kwok et al., 2005). However, ASQA (Academia Sinica Question Answering) deals with Chinese to Chinese factoid question-answering (Lee et al., 2005). It is based on hybrid architecture evaluated by six types of factoid questions: personal names, location names, organization names, artifacts, times, and numbers. ASQA combines the Machine Learning and knowledge-based approaches. It achieved respectively 37.5% and 44.5% Top1 precision for correct and unsupported answers.

Apart from Chinese and English, in NTCIR-6, Mitamura and his associates participated in four CLQA subtasks (J-J, E-J, C-C, and E-C) and introduced the Japanese language (Mitamura et al., 2007). They developped their JAVELIN system where the answers of English questions are extracted from Chinese and Japanese documents. JAVELIN III was an extension of a their previous system JAVELIN II that was initially designed to monolingual English question answering (Nyberg et al., 2005). JAVELIN III is specified by a modular, extensible and a language-independent architecture. The best run in answer precision obtained 13% for E-J and 19% for E-C.

Recently, NTCIR-9 introduced for the first time the new task RITE@NTCIR (Recognizing Inference in TExt). RITE is a generic task that handles a major understanding of the text in various research areas, such as, Information Retrieval, Question-Answering (Harabagiu and Andrew, 2006), Text Summarization, Opinion Analysis, etc. It is proposed to ensure advancements in textual inference research (Shima et al., 2011). According to these authors, RITE4QA subtask is inspired by a series of Answer Validation tasks at CLEF (Peñas et al., 2007).

2.3 Question-Answering in Arabic Language

The first Arabic question-answering system, called AQAS (Mohammed et al., 1993), has emerged in the 90s. AQAS introduces an approach based on a Human-Computer Interface. It is based on querying databases to extract the answer; transforms the question into a query to retrieve the answer and seeks answers from structured data bases that focused on a knowledge model. However, QARAB (Hammo et al., 2004) seeks answers from unstructured documents extracted from the newspaper Al-RAYA.

The majority of Arabic studies in question-answering focus on NLP techniques to extract the accurate answer, they deal with morpho-syntactic approaches. Few of these studies provide logic and inference based-approaches. In addition, ArabiQA (Benajiba et al., 2007) uses named entities techniques; QASAL (Brini et al., 2009) uses NooJ platform to identify answers from an education book. AQUASYS (Bekhti et al., 2011) uses the NLP techniques to analyze question and extract answer from an Arabic corpus and JAWEB (Kurdi et al., 2014) was constructed on the basis of AQUASYS by providing a user interface as an extension. JAWEB was exposed a formal model for a lightweight semantic-based open domain yes/no Arabic question-answering system, it is based on a paragraph retrieval (with variable length). The study of (N Bdour and Gharaibeh, 2013) proposed a constrained semantic representation using an explicit unification framework based on semantic similarities and query expansion (synonyms and antonyms). The system of Trigui and his colleges (Trigui et al., 2010), called DefArabiQA, is based on linguistic patterns, it is considered the first system provides answers to definition questions. It is based on a set of lexical patterns and uses heuristic rules to filter the candidate definitions.

Recently, Abouenour and his associates (Abouenour et al., 2012) proposed IDRAAQ system, it is based on Arabic WordNet and semantic expansion. Without using the database collections of CLEF, IDRAAQ achieved a precision 0.31 and a c@1 0, 21. ALQASIM proposed by (Ezzeldin et al., 2013) is based on selection and validation of the answer, it answers multiple choice questions. This system analysed reading test comprehension instead of questions. It achieved a performance of 0.31 precision and 0.36 c@1 without using any database collection tests.

Table 1. Comparison of arabic question-answering systems

Criteria System	Data source	Answer	Performance	Question analysis Text analysis	Question type	Features	Technique and tools used
AQAS (mohamed et al.,1993)	Structured data base	A sentence	Not Mentioned	No No	several forms (declarative statements)	Stop words Removal, tokenization	Not Mentionned
QARAB (Hammo et al., 2004)	Unstructured data base	A sentence	Precision : 97.3% Recall 97.3 %	Yes No	Question started by : من،متى، أين، كم	Type and category of expected answer.	Information retrieval IR Natural language processing NLP

ArabiQA (Benajiba et al.,2007)	Corpus	A sentence	Not Mentionned	Yes No	Factual question	Keyword and named entities recognized.	Not Mentionned
QASAL (Brini et al.,2009)	Book of education	A sentence	Not Mentionned	Yes No	Factual question	Type of expected answer, focus, keywords.	NOOJ platform
DefArabiQA (Trigui et al.,2010)	Web	A sentence	MRR: 0.81	Yes No	Definition question	Question topic, type of expected answer.	Not Mentionned
AquASys (Bekhti & Al-Harbi, 2011)	Corpus	Short answer	Precision: 66.25% Recall : 97.5% F1-Score: 87.89%	Yes No	Factual question	Keywords ; type of expected answer	Not Mentionned
IDRAAQ (Abouenour et al.,2012)	Not Mentionned	Not Mentionned	Not Mentionned	Yes Yes	QMC	Keywords ; type of expected answer	Not Mentionned
ALQASIM (Ezzeldin et al., 2013)	Not Mentionned	A sentence	Precision: 0.31 C@1: 0.36	Yes Yes	QMC	Not Mentionned	Not Mentionned
System of (NBdour and Gharaibeh, 2013)	Not Mentionned	A paragraph	Not Mentionned	Yes No	Yes/No question	Stop words	IR, NLP, artificiel intelligence (AI),
JAWEB (Kurdi et al., 2014)	Corpus	A sentence	Recall: 15-20%	Yes No	Factual question	Answer type ; Keywords	NLP

In Table 1, we present a summary of Arabic studies; the columns represent our evaluation criteria and the rows are the works studied. We conclude that the most of these studies ensure analysis of the question but do not manipulate the information associated in the text.

3 Approach

We propose to deal with the automatic text comprehension. The previous studies are based on a superficial analysis of the texts. The originality of our approach lies in the in-depth analysis of Arabic texts for the generation of textual inferences. Our method consists of five main steps, namely, analysis of the text and the question; transformation of the Arabic statements (text + question) into logical forms; the recognition of implications generated between these forms and the selection of desired answer. In the following subsections we present each one of these tasks in detail.

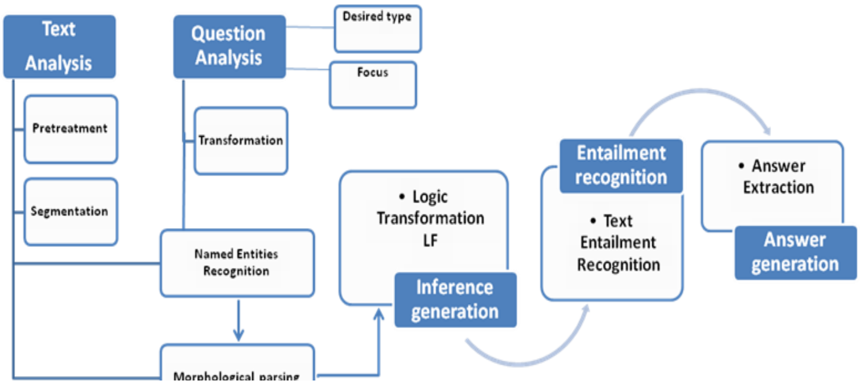
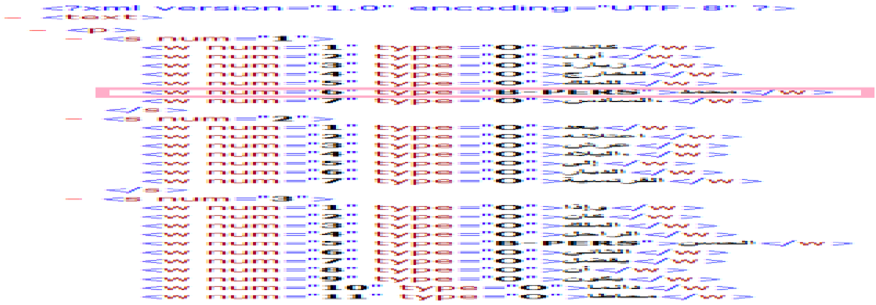


Fig. 2. Proposed approach

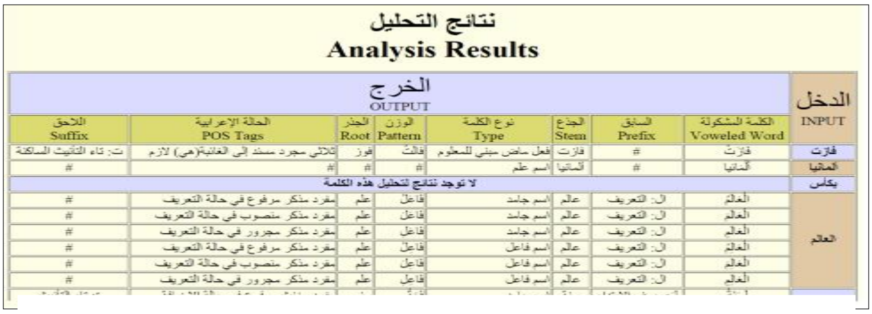
3.1 Preprocessing and Analysis

Our approach begins by an analysis step, it accepts as input an Arabic text in html format and generates an annotated and analyzed text. First, a preprocessing step is performed to clean the html text retrieved from the website "Euronews" to produce an Arabic text in txt format. Then, a segmentation step determines the division of text into tokens (sentences, words). A text analysis without segmentation lead to unreliable results (Ghassan, 2001). Afterward, the recognition of named entities is performed by ArNER (Zribi et al., 2010) to determine the set of named entities. This task is so harder in Arabic than other languages due to the lack of capital letters (Shaalan, 2014). Finally, we use Al-khalil parser to carry out the morphological analysis in order to identify the grammatical category of text words. Segmentation and Morphological analysis play a very important role in most applications of Natural Language Processing (e.g. information extraction, automatic summarization, etc.). The results of these two tools are presented successively in Figures 3 and 4.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <text>
  - <p>
    - <s num="1">
        <w num="1" type="O">[illegible]</w>
        <w num="2" type="O">[illegible]</w>
        <w num="3" type="O">[illegible]</w>
        <w num="4" type="O">[illegible]</w>
        <w num="5" type="O">[illegible]</w>
        <w num="6" type="B-PERS">[illegible]</w>
        <w num="7" type="O">[illegible]</w>
      </s>
    - <s num="2">
        <w num="1" type="O">[illegible]</w>
        <w num="2" type="O">[illegible]</w>
        <w num="3" type="O">[illegible]</w>
        <w num="4" type="O">[illegible]</w>
        <w num="5" type="O">[illegible]</w>
        <w num="6" type="O">[illegible]</w>
        <w num="7" type="O">[illegible]</w>
      </s>
    - <s num="3">
        <w num="1" type="O">[illegible]</w>
        <w num="2" type="O">[illegible]</w>
        <w num="3" type="O">[illegible]</w>
        <w num="4" type="O">[illegible]</w>
        <w num="5" type="B-PERS">[illegible]</w>
        <w num="6" type="O">[illegible]</w>
        <w num="7" type="O">[illegible]</w>
        <w num="8" type="O">[illegible]</w>
        <w num="9" type="O">[illegible]</w>
        <w num="10" type="O">[illegible]</w>
        <w num="11" type="O">[illegible]</w>
```

Fig. 3. Annotated text with ArNER



نتائج التحليل
Analysis Results

اللاحق Suffix	الحالة الإعرابية POS Tags	الجذر Root	الوزن Pattern	نوع الكلمة Type	الجذع Stem	السابق Prefix	الكلمة المشكولة Voweled Word	الدخل INPUT
				الخرج OUTPUT				
ت: تاء التأنيث الساكنة	ثلاثي مجرد مسند إلى الغائبة(هي) لازم	فوز	فالَتْ	فعل ماض مبني للمعلوم	فازت	#	فَازَتْ	فازت
#	#	#	#	اسم علم	ألمانيا	#	ألْمانيا	المانيا
لا توجد نتائج لتحليل هذه الكلمة								بكأس
#	مفرد مذكر مرفوع في حالة التعريف	علم	فاعَل	اسم جامد	عالم	ال: التعريف	العَالَم	العالم
#	مفرد مذكر منصوب في حالة التعريف	علم	فاعَل	اسم جامد	عالم	ال: التعريف	العَالَم	
#	مفرد مذكر مجرور في حالة التعريف	علم	فاعَل	اسم جامد	عالم	ال: التعريف	العَالَم	
#	مفرد مذكر مرفوع في حالة التعريف	علم	فاعِل	اسم فاعل	عالم	ال: التعريف	العَالِم	
#	مفرد مذكر منصوب في حالة التعريف	علم	فاعِل	اسم فاعل	عالم	ال: التعريف	العَالِم	
#	مفرد مذكر مجرور في حالة التعريف	علم	فاعِل	اسم فاعل	عالم	ال: التعريف	العَالِم	

Fig. 4. : Morphological parser with Al-Khalil

The second step of our approach consists in analyzing the user's question. This is a preliminary step of the answer research process. Generally, the features extracted from this step facilitate the answer extraction. In Arabic, the majority of studies focuses on the extraction of keywords and named entity recognition. In our case, we focus on the reformulation of the question into a declarative form. This must be used shortly for generating logical forms.

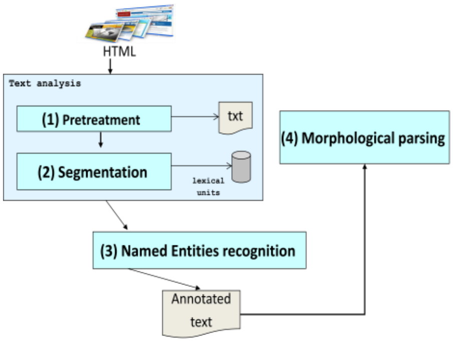


Fig. 5. Steps of text analysis

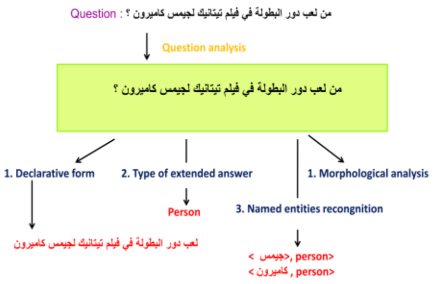

Question : من لعب دور البطولة في فيلم تيتانيك لجيمس كاميرون ؟
Question analysis
من لعب دور البطولة في فيلم تيتانيك لجيمس كاميرون ؟
1. Declarative form
2. Type of extended answer
1. Morphological analysis
Person
3. Named entities recongnition
لعب دور البطولة في فيلم تيتانيك لجيمس كاميرون
< جيمس>, person>
< كاميرون , person>

Fig. 6. Steps of question analysis

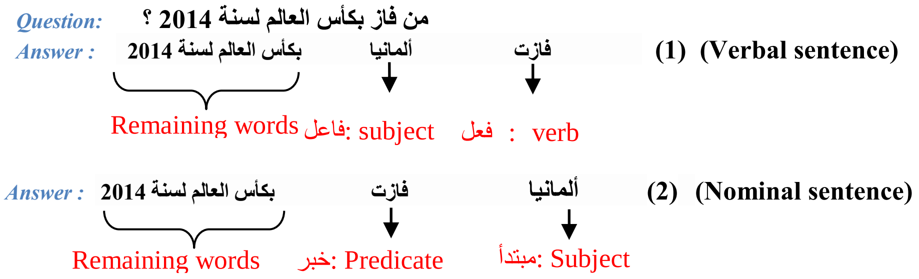
3.2 Transformation of the Question and the Text into Logic Forms

In this section, we look at the transformation of Arabic statements (question and text) into logic forms which affect the logical meaning of an Arabic sentence. Logic form is an intermediary step between the syntactic parse and the deep semantic form (Moldovan and Rus, 2001). Semantics is the most difficult level at which the language makes a contact with the real world. It is the most important part of Natural Language Processing. Practically all researches on the question answering in Arabic focus on morpho-syntactic approaches. Although, several approaches had been proposed for semantics, very few of them applied to Arabic. Logic-based approaches were applied to English and some other languages but none of them has been applied to Arabic yet.

Our approach deals with verbal and nominal simple sentences using AL-Khalil morphological parser to determine the grammatical category words in the Arabic statements. In the transformation of the Arabic statements into logical forms, we create our logic representations from a text and question data which are divided into two classes:

- A verbal sentence begins with a verb, and it has at least a verb (فعل) and a subject (فاعل). The subject can be indicated by the conjugation of the verb, and not written separately.
- A nominal sentence begins with a noun, it have two parts: a subject (مبتدأ) and a predicate (خبر). In this case, only nominal elements are used as constituents. The subject of the nominal sentence is a noun or a pronoun, while the predicate can be a noun, adjective, preposition and noun, or verb.

For a logical representation, we refer to a semantic representation of statements in natural language, especially the logical structure. A predicate expression is a graph of predicate-argument relationship, we work with the following example:


Question: من فاز بكأس العالم لسنة 2014 ؟
Answer : فازت ألمانيا بكأس العالم لسنة 2014
Remaining words
فاعل: subject
فعل : verb
(1) (Verbal sentence)
Answer : ألمانيا فازت بكأس العالم لسنة 2014
Remaining words
خبر: Predicate
مبتدأ: Subject
(2) (Nominal sentence)

Arabic is different from English and other languages in the words order, the verb and noun criteria and the sentences type. In our case, we deal with verbal and nominal sentences. In verbal sentence, we handle the transitive and intransitive verb in order to assume the logical representations. In nominal sentence, we care the common and proper nouns. The identification of such predicate depends on its context. The predictive structure is a graph of predicate-argument relations.

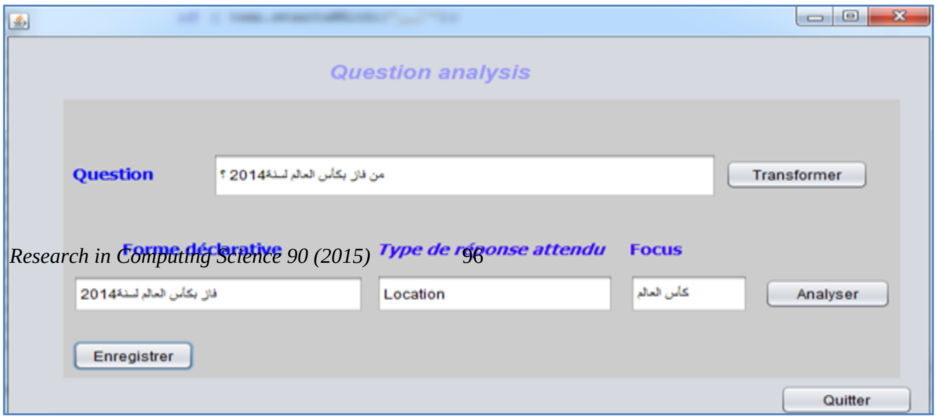
Table 2. Mapping the words of a sentence to logical forms

Sentence words	*Rules*	*Logic predicates*
Example 1:	**فازت ألمانيا بكأس العالم لسنة 2014**	**(case of a transitive verb)**
Proper noun German almania ألمانيا	***logical constant*** Germany Almania ألمانيا	Germany Almania ألمانيا
Transitive verb win faza فاز	***Two predicates*** (λY) (λX) win (X,Y) (λY) (λX) faza (X,Y) (λY) (λX) فاز (X,Y)	Y^ X^ win (X,Y) Y^ X^ faza (X,Y) Y ^X^ فاز (X,Y)
Example 2:	**نام الطفل**	**(case of an intransitive verb)**
Common noun student tilmidh تلميذ	***One predicate*** (λX) student (X) (λX) tilmidh (X) (λX) تلميذ (X)	X^ student (X) X^ tilmidh (X) X^تلميذ (X)
Intransitive verb sleep naama نام	***One predicate*** (λX) sleep (X) (λX) naama (X) (λX) نام (X)	X^ sleep (X) X^ naama (X) X^ نام (X)

3.3 Implementation of the proposed approach

We have implemented our approach with the Java language, in NetBeans. We use ArNER to determine the named entities and Al-Khalil parser to identify the grammatical category of words. The results of these two steps are grouped into XML files.

In our work, the question analysis determines the expected answer type and reformulates the question in the affirmative form in order to generate hypotheses to be passed to the logical transformation module.
Question Analysis module is shown in Fig. 7.



Question: من فاز بكأس العالم لسنة2014 ؟

Forme déclarative: فاز بكأس العالم لسنة2014

Type de réponse attendu: Location

Focus: كأس العالم

Fig. 7. Question analysis

4 Conclusion

Logic is a level between the syntactic and the deep analysis, it is the most important and difficult level in Natural Language Processing. Logic-based approaches are a rich research topic even though there is still room for improvement. This task was applied for many other languages (English, French...) but not yet to Arabic. This is due to the lack of necessary tools of Arabic and the specificities of this language. Our approach is a representative solution for Arabic.

In this paper, we proposed an original method intended for the Arabic within the framework of the automatic Arab statements comprehension to generate logical predicates. It is based on logic formulas. This method involves four steps: the textual statements analysis, the transformation of the question and text into logic forms, the textual entailments recognition and the desired answer retrieval.

For future work, we will extend this work to cover others cases in Arabic. We plan to study the phenomenon of the textual entailment recognition from logical forms generated. Then, we address these implications in the process of the desired answer selection. We investigate all cases studied logic formulas to generate the logical representations of textual statements. Also, we plan to implement this proposed model for the generation of the correct answer.

Acknowledgements

I give my sincere thanks for my collaborators Professor Patrice Bellot (University of Aix Marseille, France) and Mr Mahmoud NEJI (University of Sfax-Tunisia) and Mr Omar Trigui (University of Sfax-Tunisia) that I have benefited greatly by working with them.

References

1. Abouenour L., Bouzoubaa K. and Rosso P. (2012). "IDRAAQ: New Arabic Question Answering System Basedon Query Expansion and Passage Retrieval". *CLEF (Online Working Notes/Labs/Workshop).*
2. Athenikos SJ., Han H. (2010)."Biomedical question answering: A survey", Computer Methods and Programs in Biomedecine 99 (1):24, *PMID*, 19913938.
3. Bdour W. N and Gharaibeh N. K. (2013) "Development of Yes/No Arabic Question Answering System", *In International Journal of Artificial Intelligence & Applications (IJAIA),*. Vol.4. No.1, January 2013. DOI: 10.5121/ijaia.2013.4105 51.
4. Bekhti S., Rehman A., AL-Harbi M. and Saba T. (2011)."AQUASYS: an arabic question-answering system based on extensive question analysis and answer relevance scoring", In *International Journal of Academic Research*; Jul2011, Vol. 3 Issue 4, p45.
5. Benajiba Y., Rosso P. and Lyhyaoui A. (2007). "Implementation of the ArabiQA Question Answering System's components", In: Proc. *Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium. ICTIS-*2007, Fez, Morroco, April 3-5.

6. Bhaskar P., Pakray P., Banerjee S., Bandyopadhyay S. and Gelbukh A. (2012). “Question Answering System for QA4MRE@CLEF, *Workshop on Question Answering For Machine Reading Evaluation, QA4MRE*, September.
7. Brini W., Ellouze M., Mesfar S. and Belguith L. H. (2009)“An Arabic Question-Answering system for factoid questions”, In *IEEE International Conference on Natural Language Procesing and Knowledge Engineering (IEEE NLP-KE'09),* Dalian, China.
8. Cabrio E., Cojan J., Aprosio A-P., Magnini B., Lavelli A. and Gandon F. (2012). “QAKiS: an Open Domain QA System based on Relational Patterns”, *11th International Semantic Web Conference (ISWC 2012),* Demo paper. Boston, USA.
9. Cao Y., Liu F., Simpson P., Antieau L., Bennett A., Cimino JJ., Ely J. and Yu H. (2011) “AskHERMES: an online question answering system for complex clinical questions”, *J Biomed Inform*, 44(2):277–288.
10. De Chalendar G., Dalmas T., Elkateb-Gara F., Ferret O., Grau B., Hurault-Plantet M., Illouz G., Monceaux L., Robba I., Vilnat A. (2002). “The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet”, *TREC11*, NIST special publication SP.
11. Ezzeldin A. M., Kholief M. H. and El-Sonbaty Y. (2013). “ALQASIM: Arabic Language Question Answer Selection in Machines”, *CLEF 2013*, 2013. P 100-103.
12. Ferret O., Grau B., Hurault-Plantet M., Illouz G., Jacquemin C., Masson N. and Lecuyer P. (2000) “QALC-the question answering system of LIMSI-CNRS”, Technical report: LIMSI-CNRS, *TREC 9*, evaluation.
13. Ghassan M. (2001) Analyse informatique des signes typographiques pour la segmentation de textes et l’extraction automatique des citations. Réalisation des Applications informatiques : SegATex et CitaRE, PhD thesis, Paris-Sorbonne.
14. Grau B., Pho V-M., Ligozat A-L., Ben Abacha A., Zweigenbaum P. and Chowdhury Md.F.M. (2012). “Adaptation of LIMSI's QALC for QA4MRE”, *CLEF (Online Working Notes/Labs/Workshop).*
15. Hammo B., Ableil S., Lytinen S. and Evens M. (2004). “Experimenting with a Question Answering system for the Arabic language”, In *Computers and the Humanities*. Vol. 38, N°4. Pages 397—415.
16. Harabagiu S. and Andrew H. (2006). “Methods for using textual entailment in open-domain question answering”, In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL.
17. Kurdi H., Alkhaider S. and Alfaif N. (2014). “Development and evaluation of a web based question answering system for arabic language”, In *Computer Science & Information Technology (CS & IT)*, 04(02), 187 – 202.
18. Kwok K-L., Choi S., Dinstl N. and Deng P. (2005). “NTCIR-5 Chinese, English, Korean Cross Language Retrieval Experiments using PIRCS”, *In: Proc. of the Fifth NTCIR Workshop Meeting. NII*, Tokyo, pp.88-95.
19. Lee C-W., Shih C-W., Day M-Y., Tsai T-H., Jiang T-J., Wu C-W., Sung C-L., Chen Y-R., Wu S-H., and Hsu W-L. (2005). “Perspectives on Chinese Question Answering Systems”, *in Proceedings of the Workshop on the Sciences of the Artificial (WSA 2005)*, Hualien, Taiwan.
20. Lee C-W., Shih C-W., Day M-Y., Tsai T-H., Jiang T-J., Wu C-W., Sung C-L., Chen Y-R., Wu S-H. and Hsu W-L. (2005) “ASQA: Academia Sinica Question Answering System for NTCIR-5 CLQA”, *in Proceedings of NTCIR-5 Workshop*, Tokyo, Japan, 202-208.
21. Li X. and Croft W.B. (2001). “Evaluating Question Answering Techniques in Chinese” HLT *Notebook Proceedings*, pp. 201-206.

22. Ligozat A-L. (2013). "Classification de questions par traduction", *COnference en Recherche d'Informations et Applications, CORIA*.
23. Mitamura T., Lin F., Shima H., Wang M., Ko J., Betteridge J., Bilotti M., Schlaikjer A. and Nyberg E. (2007). "JAVELIN III: Cross-Lingual Question Answering from Japanese and Chinese Documents", *In Proceedings of NTCIR-6 Workshop*.
24. Mohammed F., Nasser K. and Harb H. (1993). "A knowledge based Arabic Question Answering system (AQAS)", *ACM SIGART Bulletin*, pp. 21-33.
25. Moldovan D.I. and Rus V. (2001). "Logic Form Transformation of WordNet and its Applicability to Question Answering", *ACL* . 394-401.
26. Nyberg E., Frederking R., Mitamura T., Bilotti M., Hannan K., Hiyakumoto L., Ko J., Lin F., Lita L., Pedro V. and Schlaikjer A. (2005). "JAVELIN I and II systems at TREC 2005", *In Proc. of TREC'05*.
27. Peñas A., Rodrigo Á., and Verdejo F. (2008). "Overview of the Answer Validation Exercise 2007", *CLEF 2007*, Lecture Notes in Computer Science LNCS 5152. Springer, Berlin.
28. Shaalan K. (2014), "A Survey of Arabic Named Entity Recognition and Classification", Computational Linguistics, 40 (2): 469-510, MIT Press, USA, 2014.
29. Shima H., Kanayama H., Lee Ch-W., Lin Ch-J., Mitamura T., Miyao Y., Shi S., Takeda K. (2011) "Overview of NTCIR-9 RITE: Recognizing Inference in TExt", *NTCIR-9*, 12/2011.
30. Sitbon L., Gillard L., Grivolla J., Bellot P., Blache P. (2006). "Vers une prédiction automatique de la difficulté d'une question en langue naturelle", *13ième conférence Traitement Automatique des Langues Naturelles (TALN)*, Louvain, Belgique, p. 337-346, 10-13 Avril.
31. Tahri A. and Tibermacine O. (2013). "DBPEDIA based factoid question answering system", *International Journal of Web & Semantic Technology (IJWesT),* Vol.4, No.3, July 2013.
32. Trigui O., Belguith L.H and Rosso P. (2010), "DefArabicQA: Arabic Definition Question Answering System", *In Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC*. Valletta, Malta. 2010.
33. Voorhees E.M. (2004). "Overview of the TREC 2004 Question Answering Track", *TREC 2004*.
34. Wren JD. (2011). "Question answering systems in biology and medicine – the time is now", *Bioinformatics*, 27(14):2025–2026.
35. Zribi I., Hammami S. M. and Belguith L. H. (2010) "L'apport d'une approche hybride pour la reconnaissance des entités nommées en langue arabe", *In TALN'2010*, Montréal, 19-23 juillet 2010 (pp. 19–23).

Towards an Hybrid Approach for Semantic Arabic Spontaneous Speech Analysis

Chahira Lhioui, Anis Zouaghi, and Mounir Zrigui,

ISIM of Medenine, Gabes University, Road Djerba, 4100 Medenine Tunisia,
ISSAT of Sousse, Sousse University, Taffala city (Ibn Khaldoun), 4003 Sousse,
FSM of Monastir, Monastir University, Avenue of the environnement 5019 Monastir,
LATICE Laboratory, ESSTT Tunis, Tunisia

Chahira_m1983@yahoo.fr, Anis.Zouaghi@gmail.com,
Mounir.Zrigui@fsm.rnu.tn,

Abstract. The automatic speech understanding aims to extract the useful meaning of the oral utterances. In this paper, we propose a hybrid original method for a robust automatic Arabic speech understanding. The proposed method combines two approaches usually used separately and not considered as complementary. This hybridization has the advantage of being robust while coping with irregularities of oral language such as the non-fixed order of words, self-corrections, repetitions, false departures which are called disfluencies. Through such a combination, we can also overcome structuring sentence complexities in Arabic language itself like the use of conditional, concession, emphatic, negation and elliptical forms. We provide, in this work a detailed description of our approach as well as results compared with several systems using different approaches separately. The observed error rates suggest that our combined approach can stand a comparison with concept spotters on larger application domains. We also present, our corpus, inspired from MEDIA and LUNA project corpora, collected with the Wizard of Oz method. This corpus deals with the touristic Arabic information and hotel reservation. The evaluation results of our hybrid spontaneous speech analysis method are very encouraging. Indeed, the obtained rate of F-Measure is 79.98%.

Keywords: Speech understanding, shallow parsing, 3-level HMM, Arabic speech processing, stochastic approach, cross-validation.

1 Introduction

The context of our work is the automatic Spoken Understanding Language (SLU) and finalized Human/Machine Communication (CHM).

Various kinds of linguistic knowledge are needed for the proper functioning of an SLU module. This linguistic knowledge can be of several types: lexical, syntactic, semantic, pragmatic and sometimes prosodic for systems taking as input a transcribed text. Many forms are offered to describe these linguistic knowledge kinds among

pp. 101–122; rec. 2015-01-25; acc. 2015-04-19

them we can mention: context-sensitive grammar [16], cases grammar [11], Hidden Markov Models [9], Neural Networks [28], N-gram language models [30], λ-calculus [32], logical [17] or Unification Grammars [1] in his various forms (UCG [19], Unification Categorial Grammar; APSG [31], Augmented Phrase Structure Grammar; LTAG [22], Lexicalised Tree Adjoining Grammar; STAG [26], Semantic Tree Association Grammar).

Moreover, we distinguish essentially in automatic SLU two types of approaches for the treatment of linguistic knowledge, which are linguistic approach [18] and stochastic approach [24]. In both cases, the statement is divided into word groups. These groups are frequently called concepts [9] [24] for stochastic approach and chunks [7] for a rule-based approach.

Whereas, these two approaches frequently used separately, enable a more or less effective understanding when dealing with oral speech. In fact, when a speaker speaks in a spontaneous way, the syntax or grammar errors are much more common in spoken than in written language. In one side, this problem is not guaranteed by a detailed linguistic approach. Besides, linguistic approach, as complete as they are, requires a grateful work to analyze corpus by experts in order to extract concept spotting and their predicates. This method is limited to specific fields using restrictive language. Thus, in the case of relatively opened field, it leads to many difficulties like portability and extension which are guaranteed by stochastic approaches.

In the other side, unexpected oral structures do not also obey to any statistical law and may not be satisfactorily modeled by a stochastic approach [23]. Besides, stochastic models do not seem to be able to solve the problem of a detailed language analysis. Added to that, the current stochastic models based on the only restricted linguistic entities observation sequences (words, syntactic categories, concepts, etc) cannot encounter statement deepness structures [23]. Finally, learning or training statistical techniques require a large amount of data annotated for learning stochastic model which is not always available.

However, it would be absurd to reject utterances which are syntactically incorrect because the goal is not to check the conformity of user’s utterances to syntactic rules but to extract rather the semantic content. Hence, semantic level is important to control meaning of utterances and both syntax and semantic should have to be controlled in order to do not cause understanding problems. Stochastic models are more permissive than linguistic one based on formal grammars. They accept all the sentences of a language. Even incorrect sentences are accepted.

Qualities and drawbacks of these two separately-used approaches allowed us to detect certain complementarities between both of them. This observation is the motivation of our work that attempts to combine the two mentioned-above approaches to take advantage of their strengths. In this regard, we have proposed an hybrid approach based on linguistic and probabilistic methods to the semantic analyzer development for standard Arabic uttered sentence. This is achieved by integrating linguistic details describing local syntactic and semantic constraints on an Hidden Markov Model (HMM) stochastic model. The later materializes the language model of our Touristic Information and Hotel Reservations (TIHR) application.

2 Related Works and Motivation

Oral utterances are so often ungrammatical or incomplete and therefore an important part of the information contained in their texts is lost during an only syntactic rule-based analysis. That is why the analysis covering only syntax aspects is generally not effective. Added to that, Automatic Speech Recognition (RAP) systems generate a significant number of errors not contoured by grammars. Thus, to deal with all these oral treatment problems, some propose either a detailed linguistic phenomena analysis such as in [3], or a combination of a syntactic and semantic analysis such as in [31]. The others resort to stochastic methods as they are more robust to the transcription errors and adapt better to the specificities of oral language. Among these works, we cite for example the work of [9] which aims at achieving a stochastic conceptual decoding based on HMM with two levels in the context of robust spontaneous speech understanding. In the same context, [25] used also a Bayesian stochastic approach to speech semantic composition in Human/Machine dialogue systems.

Unlike Latin languages, the automatic understanding of spontaneous Arabic speech using stochastic approaches receives little attention at scientific research. In our knowledge, we note that the only work related to the use of stochastic techniques for the spontaneous Arabic language understanding are those of [35]. They have used a stochastic language model for spontaneous Arabic speech semantic analysis in the context of a restricted field (Train Information).

Thus, the originality of our work is to combine two common approaches used to be treated separately (linguistic and stochastic approaches) to the development of a semantic analyzer for standard Arabic utterances. This semantic analyzer is dedicated to tourists who communicate with the Interactive Voice Server (IVS) using Standard Arabic language in order to learn about touristic information that concerns them. Its principal role is then the construction of semantic representations for their utterances.

This hybrid approach consists, in fact, on integrating linguistic constraints in stochastic HMM model. Linguistic analysis should not inhibit the understanding process on the pretext that the input data are grammatically incorrect. According to the Blache citation appeared in [7], shallow parsing is a linguistic technique which facilitates greatly language processing. It rather conforms to spoken irregularities since it is only interested in extracting of word pieces containing only useful information (which are called chunks). Hence, a shallow parsing technique was chosen in our case.

Therefore, results of the linguistic analysis are syntaxico-semantic rules that governing chunks constituting user utterances. These syntaxico-semantic rules play the role of linguistic constraints applied in the first step of our hybrid analysis strategy. The second step will obviously be a stochastic HMM-decoding and then, HMM observations will be these extracted syntaxico-semantic constraints.

The choice to start with a linguistic analysis is appreciated from the fact that the latter does not allow if statements are utterly invalid. Hence, starting with a linguistic one prevents the spread of fatal errors between analyzer modules set in pipeline (see Fig. 1.).

We also note that our resulting hybrid model shows a difference from that of Bousquet [8]. This difference consists in having a 3-level HMM instead of a 2-level

Bousquet HMM. These three levels describe respectively three data types: syntaxico-semantic rules, conceptual and probabilistic information. Known that our application domain (TIHR) is relatively open, given its richness in terms of concepts, we numbers 83 between concepts and conceptual segment (see Section 4.3) and 163 linguistic rules (see Section 4.2).

3 Difficulties in Arabic Language speech Semantic Analysis

3.1 Disfluencies in spontaneous utterances

As indicates their Etymology, disfluencies mean any interruption or disturbance of influence [10]. In what follows, we focus on the major phenomena of disfluencies i.e. self-corrections, repetitions and hesitations.

- Self-correction: this case appears where the speaker made one or more mistakes and corrects in the same statement and in the real time. In this case, the wrong word (or words) is completely pronounced [8].
- Repeating: this is the case of the repetition of a word or series of words.
- Hesitation: it is a break filled in oral production which can be manifested in various ways: by using a specific morpheme (e.g. 'آم' (Eum), 'آه' (Euh) etc.) or in the form of a syllable elongation [10].

3.2 Semantic difficulties in Arabic language analysis

Arabic statements semantic analysis is a very difficult task given its semantic richness. This complexity is due to Arabic language specifics, which are:

- Arabic word can mean an entirely expression in English or French language [7]. For example the word ″أرأيت؟″ (ara ayta) designs in English language 'Did you see?″. Thus, the automatic interpretation of such words requires a prior segmentation which is not an easy task [35].
- Words order in Arabic sentence is relatively variable compared, for example, to English or French languages where words order arrangement in a sentence respects perfectly the sequencing SVO (Subject Verb Object). Whereas, in Arabic, we always have the liberty to begin with terms in order to put the stress on. In addition, Arabic oral statements do not generally respect any grammar because of the oral spontaneous character. This complicates the task of building a grammar rules to be used in the semantic interpretation process. To do this, we must provide in such grammar all possible combination rules of words order inversions in a sentence.
- The non-capitalization of proper nouns, acronyms and abbreviations makes identification of such words even more difficult than the Latin languages.
- The connection without space of the coordination conjunction 'و' (and) to words. This makes it difficult to distinguish between 'و' as a word letter (e.g. وقف(stand)) and the 'و' having the role of a coordinating conjunction. But this type of combina-

tion plays an important role in the interpretation of a statement by identifying its proposals.

- The pronunciation nature of some Arabic letters, for example: غ(gh: ghayn pronunciation). These phonemes have no equivalent in other languages, such as, for example, French or English. Besides, some letters of the Arabic language, such as: ف(f: Fa pronunciation), ح(h: Hha pronunciation) خ(kh: Kha pronunciation) ض (d: Dad pronunciation), ذ (d: pronunciation Thal) ظ (z Zah pronunciation), are pronounced by a strong expiration; so the quality of the microphone can affect speech recognition results;
- The possibility of existence of many graphemes for the same phoneme (e.g. graphemes and ظ ض), or several realizations for the same phonetic grapheme (e.g. grapheme 'ل' (lam) has two different output sounds, depending on letters that precede and follow, as in: 'بالله' (please) and 'الله' (god)). Some graphemes may not be considered in pronunciation ('ـا' for the elongation sound). This phenomenon makes difficult a speech recognition task and, consequently, the task of understanding.

4 Our Hybrid Method Approach

To carry out a robust Arab statements semantic analysis task, we decided to go through the two following steps:

- Chunking. This step is done by using a shallow rule-based parsing. Such analysis can often make a partial parsing in order to extract only essential elements that construct statements. These elements are called the chunks [13]. This analysis is guided in our case by a syntactic-semantic Probabilistic Context-Free Grammar (PCFG). The choice of opting for a PCFG is dictated by its adaptability to oral grammatical irregularities. In fact, in ASP (Automatic Speech Processing) field, the PCFG is often used in oral treatment cases. The result of chunking step is set of local linguistic constraints which govern extracted chunks.
- Stochastic analysis. During this step, a stochastic semantic decoding module transforms linguistic constraints extracted in the previous step on concepts.

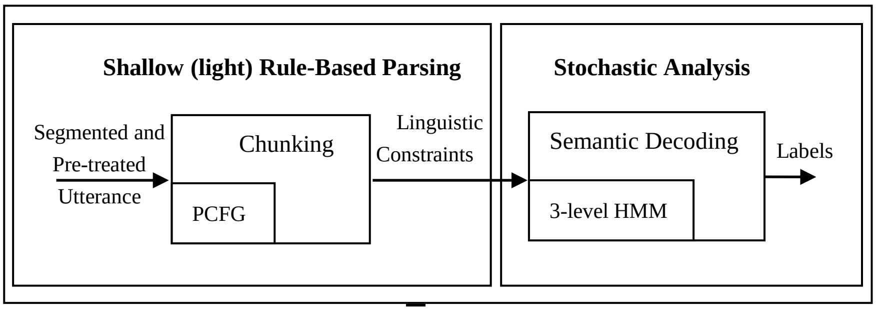


Fig. 1. The hybrid semantic analysis process

Fig. 1 describes our hybrid semantic analysis strategy adopted in Arab statements semantic analysis.

In the following, we detail each step separately.

4.1 Segmentation and Pretreatments

Because of the spontaneous utterance nature that contains various types of disfluencies (see at 3.)), an oral statement is inherently rigid and difficult to control. These disfluencies are frequent phenomena that appear normally in spontaneous speech. Here is an example of hesitation and self correction statement [12]:

هل يوجد مطعم خاص بالكباب هنا آه عفوا بالبيتزا
Is there a restaurant special kabab here, ah pizza sorry?

All these phenomena lead to the ambiguity problems. That is why a pretreatment step is required. This later removes duplication and unnecessary information, to divide complex utterances into elementary ones, to convert numbers written in all letters, and to determine the canonical forms of words [4] [14].

4.2 Chunking or Shallow rule-based Parsing

The basic idea of this analysis is to limit the depth and richness of a full parsing for the essentials. It consists on a partial parsing that promises to get minimal linguistic structures. These structures are called chunks. They are useful such as in other applications in that they promote their portability.

Our choice to use the chunk notion is based on the Blache citation which asserts that "chunks facilitate the analysis of a statement" [7]. To achieve this, we use the following chunk definition:

Definition1. A chunk is defined as a result of non-recursive categories formed of a head, to which function words and adjacent modifiers may be added [6].

According to the definition 1, we consider a chunk as typical segment consisting of a full word surrounded by functional words and adjacent modifiers. These are connected to the full word by the following linguistic rule:

Chunk_Name → NP

NP → Det (Num) (Adj) Name

where the full word appears as a Name, the functional word is the Det and adjacent modifiers are (Num) and (Adj).

Shallow rule-based parsers are often characterized by a two-step process: pattern recognition step and word sense disambiguation one. Similarly, we envisage two independent steps for the realization of our shallow parser.

- Chunk identification
- Chunk attachment (through linguistic rules)

Chunk Identification. The main problem in a shallow rule-based parsing is the recognition of chunks. To resolve it, we applied a "Pattern recognition" algorithm used by many robust parsers [13] from which they take their robustness. The main idea of our algorithm is to loop the input string n times while calculating at each time the activation chunk degree present in this statement. Given that, according to what has been described in the cognitive ACT-R theory, it is possible to characterize a chunk by its activation level. This activation is described by the following equation extracted from [7]:

$$A_{ik} = B_i + \sum_j w_j \, S_{ji} \qquad \mathbf{(1)}$$

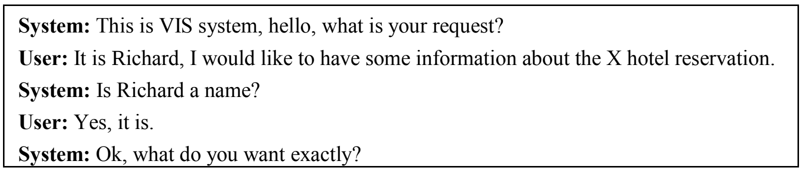
where $k <= n$. In this formula, B represents the latency degree since the last access to the chunk. It is known as the basic activation and stores the frequency and the history of this chunk access. W corresponds, to the items or terms weight which are associated to the chunk. These weights are known as the 'sources' that can activate considered chunks.

Chunk Attachment. Relationships activating a chunk can be viewed as attachments properties that we seek to maximize. Solving chunk attachment problem is reduced to the study of two attachment types: intra-chunk attachment, which concerns words order inside the chunk, and inter-chunk attachments that defines relationships between different chunks constituting statements.

Intra-Chunk Attachment. The first attachment form affects words order inside a chunk. The example below shows a syntactic ambiguity of prepositional group attachment [with mayonnaise PP] either to the verbal group [want to eat VP] or to the nominal group [kebab NP]. This syntactic ambiguity causes a semantic ambiguity since we know more then, if the kebab is mixed with mayonnaise or the man asks two separated entities; one is the kebab and the other is the mayonnaise.

أريد أكل الكباب بالمايونيز
I want to eat kebab with mayonnaise.
(the kebab and mayonnaise are baked together) or (the kebab and the mayonnaise are baked separately)

Lexical ambiguities are compounded by the attachment ambiguities, especially syntactic ambiguities. In the dialog presented in the Fig 2, the initial user statement causes an analyzing system problem because, according to his knowledge, the lexical item "Richard" can be categorized either as a noun or as a surname. The rest of the utterance is analyzed without difficulty.



System: This is VIS system, hello, what is your request?
User: It is Richard, I would like to have some information about the X hotel reservation.
System: Is Richard a name?
User: Yes, it is.
System: Ok, what do you want exactly?

Fig. 2. A lexical ambiguity resolution

Regarding the resolution of lexical ambiguities, it can be assigned to the system in order to create an action. This action gives a communicative purpose that leads to the statement "Is 'Richard' a name?". The user confirmation of this hypothesis allows the system to finish the analysis of the initial user statement, and the search in the user's message box. Then we can say here that the interaction is closed by the user.

However, semantic ambiguities can occur even as in expressions where there are no syntactic or lexical ambiguities. For example, "the coast road" can be the road that follows the coast or the road that leads to it.

One example of syntaxico-semantic intra-chunk ambiguities are uses of condition constraints when the condition chunk ‘لا.... إلا’ (<not> ... <only>) is scattered on two fragments: الشرط and جواب الشرط where ' الشرط ' (the proposal) is the portion that lies between locution fragments <not> ... <only> and ' جواب الشرط ' (the proposal response) is the portion that comes after <only>. The example below illustrates what we are trying to explain:

–لا أريد السفر إلا على متن طائرة
-I do not want to travel only by a plane

where the use of the negation form in the proposal leads to a positive form in the proposal response to emphasize it. This complex formulation can easily and successfully be resolved by linguistic rules.

Inter-Chunk Attachment. The second form of chunks attachment concerns, in this case, the rearrangement of chunks within the same proposal. This problem increases with the Arabic language where words order is generally not fixed (see at 3.2). In fact, the sentence in the Arabic language do not undergo the SVO (Subject, Verb, Object) form, as it is the case of French or English languages. We cite two main categories which are the basis for the construction of Arabic[1] sentences: the first category is SVO (Subject Verb Object) that concerns nominal sentences, while the second, VSO (Verb Subject Object), is a form that describes verbal phrases. This is particularly critical in the case of Arabic language since it is morphologically rich.

– سوسة مدينة ساحلية(جملة إسمية)
- Sousse is a coastal city (Nominal sentence)
– أريد السفر إلى سوسة(جملة فعلية)
- I want to go to Sousse (Verbal sentence)

[1] The problem is the same for other source languages such as Chinese or Russian, and the solutions proposed here will therefore apply in other contexts.

Several tricks can also inhibit an automatic understanding of this language. Among them we can mention: reversals (تقديم وتأخير) of sentence constituents as it is the case of nominal sentences where the proposal (الخبر), putted as a prepositional phrase, may precede the theme (المبتدأ). The example below illustrates this reversal that results after the preposition placement of (حرف الجر) 'إلى' putted at the beginning of the subject:

– أريد السفر إلى X
- I want to travel to X
– إلى Xأريد السفر
- To X I want to travel

A Constraints Grammar. A solution adapted to the intra and inter-chunk attachment problem is the description of different attachment constraints by a set of linguistic rules. This set is called "selection constraints". In this regard, we have designed and specified a probabilistic constraints grammar (see table below) with a probability distribution on the set of potentially finite rules defined by:

$$p(t) \geq 0$$
$$\sum_{c \in T_G} p(c) = 1$$

where t denotes the derivation tree which generates the constraint selection $c \in V^*$ and T_G designs all the derivation trees with G is a grammar describing all these constraints.

Table 1. Definition of probabilistic constraints grammar

Sets	Description	Statistics
N : Set of non-terminal symbols	Syntactic, semantic and lexical terms : SN, SV, ADJ, VDestination, etc.	91
V : Set of terminal symbols	Linguistic constraints: e.g. C_El_Jar, cause/consequence, condition, WaElMaîya constraint.	43
S : Axiom	Chunk_attach	-
R : Set of production rules	The set of rules r_i linking the constraints	163
$\mathbf{p_i}$ **:** Probability attributed to the r_i rule	A positive coefficient attributed to each rule r_i	-

The elaboration of grammar constraints consists in the definition of:

- The inter-chunk syntax: this is equivalent to indicate how chunks can be arranged together. In our case, this information is modeled by bi-gram transitions that relies different chunks on each other.

- The intra-chunk syntax: this concerns words ordering within each chunk. Each chunk is also modeled by an HMM with bi-grams transition probabilities which relies all words that form the chunk.

For example, the linguistic selection constraint characterizing the condition phenomenon and connecting two chunks forming the condition rule is:

Chunk_condition → not C1 only C2
C1 → الشرط
C2 → جواب الشرط

Thus, the probabilistic specified constraints grammar can be represented as a 2-level HMM as described in the following figure:

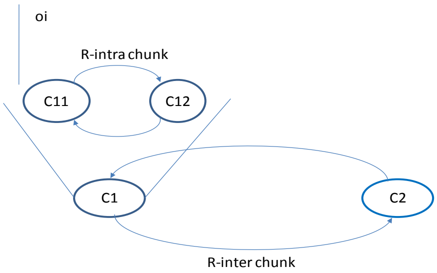


Fig. 3. The representation of constraint grammar on a 2-level HMM

4.3 Stochastic Analysis

Our stochastic analysis is based on concepts and conceptual segments notions. Their definitions are inspired from [9]. To achieve this, we used these two following definitions:

Definition 2. A concept (C) is defined as "a general and abstract mental representation of an object", and is independent of the language [8].

Definition 3. A conceptual segment (SC) comprises word sequences expressing the same meaning unit.

A word sequence making a conceptual segment is an instance of this conceptual segment. For example, the two following word sequences "at five o'clock" and "about four or five o’clock" are two instances of the Time conceptual segment.

According to definition 2. and definition 3., we conclude as a corollary 1. that a concept can be instantiated by several conceptual segments. In addition, according to the definition 1. and definition 3., we conclude as a corollary 2. that a conceptual segment can be formed by one or more disjoint chunks.

Example:

أريد[الذهاب /إلى سوسة]
I want [to travel \ to Sousse]

الذهاب (to travel) and إلى سوسة (to Sousse) chunks form the conceptual Travel segment.

Statement Semantic Representation Principles.

The user statement E is composed of n sequence words noted m_i.

$$E = m_1 m_2 \dots m_n$$

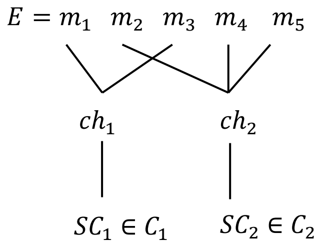
Then, we suppose the following hypothesis: Any statement E expresses a suite of k concepts. These concepts are also expressed using a suite of l conceptual segments and each conceptual segment is formed by a sequence of j chunks.

$$C = C_1 C_2 \dots C_n$$
$$SC = SC_1 SC_2 \dots SC_l$$
$$SC_i = ch_1 ch_2 \dots ch_j$$

The statement E is linearly decomposed into series of l conceptual segments, and hence of $j \times l$ suites of chunks.

$$E = SC_1 SC_2 \dots SC_l = (ch_j) \dots (ch_j)_l$$

For example, if the statement E has five words denoted m_i $(1 \leq i \leq 5)$, and each m_i belongs to a chunk ch_k (1 ≤ k ≤ 2), and if E is divided into two conceptual segments SC_1 and SC_2 where each one belongs to a proper concept, a possible representation of the statement E is:



The principle of the statement interpretation with this representation is reduced then to divide into chunks, SC and concepts. This modeling with three levels of knowledge (chunks, SC and concepts) can be represented using a 3-level HMM.

Determination of Concepts and Conceptual Segments. The creation of a model language of a given application requires the determination of all concepts and SC that exist in this application domain. Detailed analysis of the corpus allows determining how these concepts are expressed in his different statements and what their corresponding SC refers to.

However, the relationship between concepts and SC is not always so obvious. In fact, many possibilities are offered to the application designer. Added to that, SC can overlap and belong to multiple concepts. We postulate that the meaning of a word depends on the SC where it is located. For example, consider the following two statements:

مدينة الانطلاق سوسة

"Departing from Sousse" (A)

في اتجاه سوسة

"On the way to Sousse" (B)

The word سوسة (Sousse) exists in both statements but should be interpreted as a departure city in the statement (A) and as a destination city in the statement (B). So, according to the conceptual modeling principles of these statements, سوسة (Sousse) belongs to Departure conceptual segment in the statement (A) and to a Destination conceptual segment in the statement (B).

Another example that shows that the meaning of a word depends on the SC to which it is located. Consider the following statement:

العاشرة و عشرة دقائق من فضلك

"A ten and ten minutes please" (C)

The two numbers, in the statement (C), belong to the same concept Time and yet they must be interpreted differently. Indeed, the first number belongs to the Hour conceptual segment and the second belongs to Minute one.

Similarly, the interpretation of SC that compose the statement may depend on one other. Consider the following statement:

$$\underbrace{\text{لا أريد}}_{\text{الرفض}}\ \underbrace{\text{سوسة مدينة الانطلاق}}_{\text{الانطلاق}}$$

$$\underbrace{\text{"I do not want}}_{\text{Refuse}}\ \underbrace{\text{Sousse as a starting city"}}_{\text{Departure}}\ \text{(D)}$$

The interpretation of chunks belonging to the Departure conceptual segment depends on chunks presence in the first SC.

Principles that we select for word interpretations in the context of the statement are:

- Word interpretation depends on the chunk where it is located;
- Chunk interpretation depends on the SC where it is located;
- SC interpretation of a SC depends on the concept from which it is issued;
- Word interpretation can be modified by the presence of previous or next chunks in the statement.

Semantic Analysis Principal. Semantic analysis principles are useful to interpret every word that composes statements according to chunks and SC to which they were assigned. The interpretation may eventually be modified later by the presence of previous or subsequent chunks.

The two tables below illustrate, in terms of concepts, SC and chunks, the conceptual domain of our application that deals with Tourist Information and Hotel Reservations (TIHR) (see at 5.1).

Table 2. Statistics of concepts, SC and linguistic constraints in our 3-level HMM

Concept (1-Level HMM)	**Conceptual Segment (2-Level HMM)**	**Constraint rule (3-Level HMM)**
3	80	163

Table 3. Identification of concepts, several SC and their corresponding chunks for the TIHR field

Concepts	**Conceptual Segments (SC)**	**Chunks**
Opening_Closing_Dialog	Request	{« أريد » (I want), « أطلب » (I demand), « أرغب » (I desire), …}
	Accept	{« أقبل » (I accept), …}
	Refuse	{« أرفض » (I do not accept), …}
Hotel_Reservation	Hotel	{« نزل » (hotel), « فندق » (hostel), …}
	Room	{« بيت » (room), …}
	Tariff	{« كلفة » (cost), « ثمن » (price), « سعر » (quote), …}
	Number	{« عدد », …}
Touristic_Acknowledgement	Living_City	{« مدينة » (city), « بلدة » (town), « قرية » (village), …}
	Itinerary	{« طريق » (road), « الثنيا » (route), …}
	Transport_Mean	{« وسيلة النقل » (mean of transport), …}
	Price_Ticket	{« ثمن التذكرة », …}

Word interpretation depends on the SC where it is located follows the two principles below:

- Word interpretation depends on the concept that issued it;
- Word interpretation may be modified by the presence of a previous conceptual segment or according to the statement.

Semantic statement interpretation principles are useful to interpret every component in this statement according to word classes and SC to which it was assigned at the conceptual decomposition. The interpretation may possibly be changed later by the presence of previous or subsequent conceptual segments.

The 3-Level HMM Elaboration. The statement semantic analysis requires linguistic skills that are represented by a language model. According to our hybrid approach, our language model is defined in terms of concepts, SC and chunks attachment rules and hence modeled by a 3-level HMM. The observations of the model will be chunks simulated by linguistic constraints. These later were extracted by shallow parsing method based on PCFG analysis realized in the former step (see at 4.).

HMM Basic Principle. HMM models are powerful statistical tools that have been successfully used in various fields such as Automatic Speech Recognition (ASR) and Dialog Management (DM). They allow to model observation sequences putted in their two different forms; discreet and continuous form. Thus, we chose to represent our language model using an HMM model.

HMM are also N-grams language models. That is mean that the symbol emission probability depends on N-1 previous symbols. In our case, we consider that our HMM is a bi-gram language model.

The HMM model topology must be designed by a field expert of the considered application. Nevertheless, HMM parameters are automatically learned from training data. Therefore, stochastic models are more portable than linguistic ones where all rules must be explicitly written.

A 1-level HMM consists on two processes: the first is observable and the second is hidden. In all cases, following hidden states forms a Markov chain of order 1. A multi-level HMM is a model where observations of each hidden state are also modeled by a Markov model. The embedded Markov models number defines the model level number.

Retained Modeling. To be able to represent all the information that our language model carries out (linguistic rules, concepts and SC), a 1-level HMM is insufficient. We propose then a 3-level model (see Figure below) where: the first level is described by an HMM whose states represent our application concepts. According to the Table2. we have in total 3 principal concepts so, they correspond to three 1-level hidden states. Each concept is represented, in its turns, by an HMM whose states correspond to the conceptual segments (SC_i see Table 2.). Each SC will be represented by an HMM describing linguistic constraints of its realization.

Fig 4. shows an overview of our 3-level designed HMM.

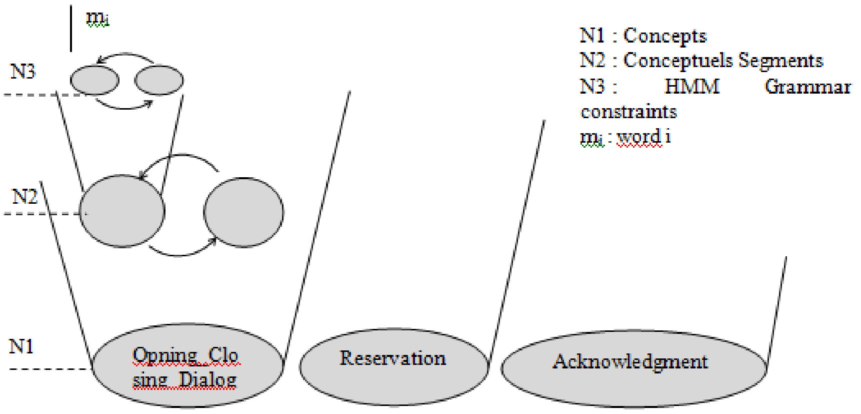


Fig. 4. Overview of our 3-level HMM

5 The Finalized Considered Application

To test our application and estimate the HMM parameters, we use a corpus dedicated to the study of touristic applications accessing to databases. The main reason for choosing this application field is the statistically representative size of the training corpus that we have. Moreover, through this corpus, we have the opportunity to produce an Arabic dialogue corpus in the same manner of those which are produced within the Francophone and Anglo-Saxon projects such as MEDIA [15] and LUNA [24]. Table 4. below shows a comparison between our Arabic dialog corpus and some other oral corpora issued in other fields and in other languages.

Table 4. Comparison between several International corpora and the our TIHR Corpus

Corpus	Language	Field	Size (on statement)
MEDIA	French	Touristic Reservation	18k
LUNA	Polonais	Transport Information	12k
TELDIR	Allemand	Time Train	22k
ATIS	English	Plan Ticket Reservation	6K
PlanRest	French	Restaurant Reservation	12k
TIHR Corpus	Arabic	TIHR	35k

5.1 Corpus Establishment

Oral corpora represent a significant proportion in the development of the automatic Spoken Language Understanding (SLU). These are closely linked to the availability of such corpus. However, Arabic speech corpora that deal with ‘Touristic Information and Hotel Reservation’ are very rare or even unavailable. Thus, to carry out our experiments in the context of this work, we were to build our own study corpus.

Our corpus is derived from the simulation of tourist information server and hotel reservations. Dialogues enunciated by tourists are in Standard Arabic Modern language (ASM). In fact, we consider that tourists are people who do not have Arabic as a native language. That is why their conversations were be uttered in the ASM language and not in common Arabic parlance.

These dialogues are about different themes such as: choice of living city, finding a route or a tourist event, a satisfaction of a price or date constraint. They were held between humans and machines through the Wizard of Oz protocol (Wizard of Oz, WoZ). Indeed, during the exchange, users believe converse with a machine while the dialogue is actually supported by a human operator that simulates informations and reservation server responses. The operator is assisted by the WOZ tool in the generation of responses to provide the user. After each user sentences, the operator shall consult the WOZ tool that offers to provide the answer to reflect the new dialogue state. To diversify the operator's responses, the WOZ tool is set at the messages, instructions and scenarios. A message sets are associated with the application to vary response formulations. With each call, the operator must follow instruction series (e.g. pretending not to have understood users to simulate mistakes which would make a real system). These instructions must be provided to the WOZ tool and depend on the scenario chosen for the dialogue recording [12].

A detailed description of technical corpus characteristics is given in Table 5.

Table 5. Characteristics of our corpus (TIHR Corpus)

	Dialog Number	**Speaker Number**	**Average Utterance Number / Dialog**	**Word Total Number**
TIHR Corpus	10 000	1000	10	730 000

All queries corpus was recorded and transcribed manually as transcription standards in XML files, and labeled according to standards proposed by the ARPA community. Manual transcription was made a loyal way that it was recorded. That is to say, words are transcribed as we hear in recordings. Hence, we note disfluencies presence such as hesitations, self-corrections, repetitions, fault departures, etc.

In our corpus, we distinguish three query types i.e. independent query context, dependent query context, and absurd requests. The study of the corpus has allowed us to identify three concepts namely, Opening_Closing_Dialog, Hotel_Reservation and Touristic_Acknowledgement (see at Table 3.). In addition, we have defined SC among that we can mention: Living_City, Hotel, Travel, Tariff_Travel, Time_Travel, Price_Ticket, Period_Travel, Itinerary, Transport_Mean, Departure_City, Arrival_City, etc. Each conceptual segment is associated with a chunks set containing reference words relating to our application field. Words constituting chunks are linked by linguistic rules governing intra-and inter-chunk relations (see at Table 1.).

5.2 Tests and Evaluation

To avoid error propagation of the former linguistic analysis (surface analysis) to the second one (stochastic analysis), manual chunk verifications and their produced rules is performed just after the linguistic analysis to evaluate the shallow rule-based parsing results. An automation of this verification is suggested as one perspective. For the evaluation of the stochastic analysis, Kohavi cross-validation [28] is performed. In fact, this latter is used to validate an existing prediction or classification model or to find the best model by estimating its precision. It helps with a reduced bias to estimate the efficiency measure of our 3-level HMM. This measure is the average of each embedded HMM. The cross-validation technique is useful when the number of observations is fixed.

Cross-validation methods principle.

The cross-validation method principle is to divide the observation set into two separate and independent subsets where the first is called training set (that is the greater one) and the second is called validation or test set. The training set is used to generate the appropriate probabilistic HMM model. Nevertheless, the test set is used to evaluate the trained model according to the evaluation criteria.

Cross-Validation Initial Conditions. To perform cross-validation test, three initial conditions have to be satisfied. Indeed, we have to:

- prepare a sample observation
- have prediction model
- mesure the method performance for two separate sets either by calculating the rate error or by measuring the efficiency, precision, recall, F-score.

Cross-Validation Method Choices. Various cross-validation methods exist. Among them we can mention three principal ones that are respectively called: Holdout (also called Test set), Leave-one-out and k-fold. A detailed description of their advantages and disadvantages of each method is found in [5]. Therefore, we have not applied the Holdout method since it does not give good results because the observation set may be small for it. This badly affects the precision of performance values measured for model evaluation. In addition, the method of Leave-out-has been abandoned in our evaluation task because it is the slowest and the less used method. So we are simply satisfied to use just the K-fold method whose principle is to find the compromise of the two pre-cited ones.

K-Fold Method. Knowing that the parameter k often used in practice is k =5 or k =10 [21], the 5-fold and 10-fold are the two respective training methods which are used in our experiments. In both cases, we have divided the observation set in k disjoint partitions of equal size where $k - 1$ partitions are used for training and the left partition is used for the test. This process is repeated such as each partition is used one time for the evaluation. We then obtain k models and therefore, k accuracy measures. The final estimated accuracy value of the model learning from the training set is the average value of k calculated accuracies.

The Model Learning Step. In this step, the Baum-Welch learning algorithm uses the training observations and an initial model for generating a decoding model. Learning is to adjust the initial model parameters. The library used in our experiments is Jahmm (JAva HMM implementation) [37]. The latter allows the HMM implementation in Java and contains the basic algorithms for using HMM. To estimate the established HMM efficiency, we used the cross-validation method. This latter allows estimating the classification model effectiveness in general and for HMM models in particular.

Initial Hypothesis. Initial parameters choice for learning or training an HMM presents some difficulties [27]. However, in our case, we treat only discrete symbols. Thus, any observation probabilities distribution is applicable to our model.

Knowing that we have a 3-level HMM, where each level L_i $(1 \leq i \leq 3)$ is characterized by n_i hidden states, we considered $1 + n_1 + n_2$ initial $HMM_{ij}(1 \leq j \leq n_i)$ which designate the i^th level and the j^th hidden state HMM. The characteristics of several models are shown in the following table. These models are classified as follow:

- HMM_1 for ($n_1 = 3$) concepts in the first level of the 3-level global HMM;
- HMM_{21}, HMM_{22}, HMM_{23} for (n_2 =80) conceptual segments in the second level. In this level we have ($n_1 = 3$) embedded HMM, where n_1 is the state number of the first global HMM level;
- ($n_2 = 80$) HMMs for n_3 =163 linguistic constraints in the third level. This number is equal to the SC number that the second level contains.

Table 6. Initial models probability distribution features (for the first and the second global HMM level)

	HMM₁	**HMM₂₁**	**HMM₂₂**	**HMM₂₃**
$\boldsymbol{\pi_i}$	Non-uniform	Uniform	Randomly	Non-uniform
$\boldsymbol{A_{ij}}$	Non-uniform	Non-uniform	Randomly	Uniform
$\boldsymbol{B_i(o_t)}$	Uniform	Uniform	Randomly	Non-uniform

The Learned Models. After the learning algorithm execution, the initial models parameters are gradually refined into a finite iteration number. We empirically chose 10 iterations. The latter value corresponds to learning algorithm stopping criterion. Following tables show average values of HMM initial probabilities simulated as two cases of k-fold cross-validation method applied for the 1-level HMM.

Table 7. Initial probabilities average value of the first level HMM with 5-fold cross-validation method

	$\boldsymbol{\pi_1}$	$\boldsymbol{\pi_2}$	$\boldsymbol{\pi_3}$	$\boldsymbol{\pi_4}$	$\boldsymbol{\pi_5}$
HMM₁	0.2	0.3	0.3	0.15	0.05

Table 8. Initial probabilities average value of the first level HMM with 10-fold cross-validation method

	$\boldsymbol{\pi_1}$	$\boldsymbol{\pi_2}$	$\boldsymbol{\pi_3}$	$\boldsymbol{\pi_4}$	$\boldsymbol{\pi_5}$	$\boldsymbol{\pi_6}$	$\boldsymbol{\pi_7}$	$\boldsymbol{\pi_8}$	$\boldsymbol{\pi_9}$	$\boldsymbol{\pi_{10}}$
HMM₁	0.1	0.03	0.09	0.07	0.17	0.11	0.25	0.08	0.02	0.08

5.3 Evaluation

In order to calculate the performance criteria and thus to assess the decoded utterance quality, we generate an n-classes confusion-matrix (Table 3). It is obtained by comparing predicted labels sequences (resulting states of the decoding step) and real labels sequence (extracted manually). Real labels sequence is determined by confusion-matrix rows information. As for predicted labels, they correspond to the confusion-matrix columns information (see Table 9).

Diagonal confusion-matrix C_i^j cells represent correct predicted labels. They correspond to the C_i label occurrence number in two state sequences. Besides, confusion-matrix C_i^j cells represent incorrect predicted labels with *i* ≠ *j*. They correspond to real and predicted C_i and C_j label occurrence numbers.

Result Interpretation. To evaluate the learning generated models, we have produced the most likely predicted states sequence (generated by the learned HMM) associated to a given observation sequence (sequence of real states). This is done for all test set observation sequences.

Subsequently, we built confusion matrices from predicted and real state sequences according to the chosen cross-validation method. Specifically, we have obtained five confusion-matrices using the K=5 method and ten confusion-matrices using K=10.

To illustrate this, we present a confusion-matrix for three labels representing the 1-level HMM states. This matrix has been obtained with the method K=5 on its first test set.

Found: L1 L3L1L2L1L3L3L2L3L3
Predicted: L1L3L1L1L1L3L3L3L3L3

Table 9. Example of a confusion matrix to 3 classes

		Prédites		
		L1	**L2**	**L3**
Réelles	**L1**	**3**	0	0
	L2	1	**0**	1
	L3	0	0	**5**

In the above-shown confusion-matrix, all label L1 predictions are correct. They are equal to three. For label L2, there is no correct prediction. On the other side, for label L3, five predicted labels are correct. We have in total eight correct labels among ten ones. We conclude then that the model generates a satisfying correct prediction numbers. Therefore, the decoding quality is satisfactory.

Performance Criteria. Table 10 presents precision, recall, and F-score average values for HMM_1, HMM_{21}, HMM_{22} and HMM_{23} embedded trained models of our 3-level HMM using K5, K10 methods. These metrics were calculated from confusion matrices for each test set.

Table 10. Precision, Recall and F-score average values for some embedded trained models using K5, K10 methods.

Modèles	Précision	Rappel	F-score
HMM_1	70.00%	71.00 %	73.79%
HMM_{21}	71.08%	68.99%	74.1%
HMM_{22}	69.98%	70.89%	72.9%
HMM_{23}	70.02%	72%	73.77%

Results Comparison. Comparison results between the Bousquet's CACAO stochastic conceptual decoder [9] and the Zouaghi and Zrigui's semantic decoder [36] shows that the error response rate, obtained by our system, was reached 20.02% which is considerably less than Bousquet system (see Table 11).

Table 11. Comparison of our hybrid system results with several well-known systems

	CACAO	**Oréodule Project decoder**	**Our analyzer**
Approach type	stochastic	semantic	Hybrid
% Error rate	29.11%	29%	20.02%

6 Conclusion and Perspectives

One of most supervised objectives that we hope to achieve when we implemented our semantic analyzer system was to fulfill spontaneous spoken Arabic language robust parsing while making the application field wider than it is currently done. Linguistic approaches are not usually viewed as efficient tools for pragmatic applications. That's why we were interesting in combining two frequently separately-used approaches (linguistic and stochastic approaches).

A second objective was to have rather a generic system, despite a field-based linguistic knowledge use. This constraint is achieved through generic rule definitions as well as their probabilities for the third linguistic integrated HMM level training. This makes it possible to estimate efficiently its parameters. Our analyzer performances show that the two divergent approaches combination can bear comparison with systems which are based on a lonely approach (stochastic approach for CACAO Bousquet's conceptual decoder and semantic approach for Zouaghi and Zrigui semantic decoder).

The model was trained using the Baum-Welch algorithm. This adjusts the initial model parameters. The learning step was carried out according to different cross-validation techniques. Then, the model evaluation was also carried out in the quantitatively and well-known methods using different metrics such as precision, recall and F-score.

As a perspective, we hope to refine our system analysis to be tested in a second challenge by evaluation campaign that will focus on the phenomena described in the typology that we have proposed.

References

1. Anne Abeillé, *Les Nouvelles Syntaxes: grammaires d'unification et analyse du français* (Linguistique). Paris: Armand Colin, 326 pp. 2 200 21096 5, 1993
2. Antoine Jean-Yves, Goulian Jérôme, Villaneau Jeane, Quand le TAL robuste s'attaque au TAL parlé : analyse incrémentale pour la compréhension de la parole spontané, TALN 2003, Batz-sur-Mer, 11-14 Juin 2003
3. Antoine, J.-Y., "Pour une Ingénierie des Langues plus Linguistique", *HDR Computer Science*, University of South Bretagne, Vannes, France, 2003
4. Bahou, Y., Belguith, H.L., Ben Hamadou, A.: Towards a Human-Machine Spoken Dialogue in Arabic. In: 6th Language Resources and Evaluation Conference (LREC

2008),Workshop HLT Within the Arabic World. Arabic Language and Local Languages Processing Status Updates and Prospects, Marrakech, Morocco, 2008b

5. Besbes G., Modélisation De Dialogues A L'aide D'un Modèle Markovien Caché Mémoire Présenté, Mémoire présenté à la Faculté des études supérieures de l'Université Laval , 2010
6. Bird S., Klein E. et Loper E., Natural Language Processing with Python, O'Reilly Media, 2009
7. Blache Philippe, Chunks et activation, un modèle de facilitation du traitement linguistique, TALN-RÉCITAL, 17-21 Juin, Les Sables d'Olonne ,2013
8. Bouraoui J.-L., Traitement automatique de dysfluences dans un corpus linguistiquement constraint, *Actes de la 16ème Conférence sur le Traitement Automatique des Langues Naturelles*, *TALN'09,* Senlis, France, 2009
9. Bousquet-Vernhettes C., Compréhension robuste de la parole spontanée -Décodage conceptuel stochastique, Thèse de doctorat, Université Paul Sabatier, 2002
10. Bove R. A Tagged Corpus-Based Study for Repeats and Self Repairs Detection in French Transcribed Speech, Proceedings of the 11 th International Conference on Text, Speech and Dialogue, TSD'08, Brno, Czech Republic., 2008
11. Bruce, B., "Cases Systems for Natural Languages", *Artificial Intelligence*, Volume 6, pp. 327-360, (1975)
12. Chahira L., Anis Z., Mounir Z., A combined A Combined Method Based on Stochastic and Linguistic Paradigm for the Understanding of Arabic Spontaneous Utterances, A. Gelbukh (Ed.): CICLing 2013, Part II, LNCS 7817, pp. 549–558, 2013.
13. François Trouilleux, Un analyseur de surface non déterministe pour le français, TALN 2009, Senlis, 24–26 juin 2009
14. Hadrich Belguith, L., Bahou, Y., Ben Hamadou, A.: Une méthode guidée par la sémantique pour la compréhension automatique des énoncés oraux arabes. International Journal of Information Sciences for Decision Making (ISDM), Septembre 2009
15. H.BonneauMaynnard, K. Mctait, D. Mostefa, L. Devillers, S. Rosset, P.Paroubek, C. Bousquet, K. Choukri, J. Goulian, JY Antoine, F. Béchet, O. Bontron, L. Charnay, L. Romary, N.Vergnes, N. Vigourous, Constitution d'un corpus de dialogue oral pour l'évaluation automatique de la compréhension hors et en contexte du dialogue, Actes des Journées d'Etude sur la Parole (JEP 2004), Fès, Maroc, avril 2004.
16. Jean-Michel Autebert, Jean Berstel et Luc Boasson, *« Context-free languages and pushdown automata »*, dans G. Rozenberg, A. Salomaa (éditeurs), *Handbook of Formal Languages*, vol. 1 : *Word, Language, Grammar*, Springer Verlag, 1997 (ISBN 978-3540604204)
17. Jeanne Villaneau, Olivier Ridoux, Jean-Yves Antoine, Comprehension de l'oral spontané Présentation et évaluation des bases formelles de LOGUS RSTI RIA.Volume18– n° 5-6/2004, pages 709 à 742, 2004
18. J. Allen., Natural language understanding. Redwood City, CA, USA : Benjamin-Cummings Publishing Co., Inc. 28, 1988
19. Karine Kray-Baschung, Gabriel G. Bes, Thierry Guillotin, French Unification Categorial Grammars https://halshs.archives-ouvertes.fr/halshs 00371433 , 2014
20. Kurdi, M-Z., Contribution à l'analyse du langage oral spontané, Thèse de l'Université Joseph Fourier, Grenoble, 2003
21. Liu, B., Web Data Mining - Exploring Hyperlinks, Contents and Usage Data, Springer, 2006.
22. Lopez P., "Analyse guidée par la connexité de TAG lexicalisées", TALN'98, Paris, 1998

23. Marcel Cori, Des Méthodes De Traitement Automatique Aux Linguistiques Fondées Sur Les Corpus, Langages, n° 171, p. 95-110. DOI : 10.3917/lang.171.0095, 2008.
24. Marie-Jean Meurs, Approche stochastique bayésienne de la composition sémantique pour les modules de compréhension automatique de la parole dans les systèmes de dialogue homme-machine, THÈSE présentée à l'Université d'Avignon et des Pays de Vaucluse, 2009
25. Meurs M., Lefèvre F. et De Mori R. , Spoken Language Interpretation: On the Use of Dynamic Bayesian Networks for Semantic Composition, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2009
26. Mohamed-Zakaria Kurdi, « A spoken language understanding approach which combines the parsing robustness with the interpretation deepness », International Conference on Artificial Intelligence (IC-AI'01), Las-Vegas, États-Unis, 2001
27. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.
28. R. Kohavir., A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, pp. 1137-1143, 1995
29. Salma Jamousi, Kamel Smaili et Jean-Paul Haton, « Contribution à la compréhension de la parole par des réseaux neuronaux », Quatrième Rencontres Jeunes Chercheurs en Parole (RJC'01), Mons, Belgique, p. 70-73, 2001.
30. S. Knight, G. Gorell, M. Rayner, D. Milward, R. Koeling, I. Lewin. Comparing grammar-based and robust approaches to speech understanding: a case study. *Proceedings of European conference on speech communication and technology*, 2001.
31. Stephanie Seneff, « TINA: A Natural Language System for Spoken Language Applications », Computational Linguistic, p. 61-86, 1992.
32. Villaneau, J., Antoine, J.-Y., and Ridoux, O., Logical approach to natural language understanding in a spoken dialogue system. In Sojka, P., Kopecek, I., and Pala, K., editors, The International Conference on Text, Speech and Dialogue (TSD), volume 3206 of Lecture Notes in Computer Science, pages 637–644. Springer, 2004
33. Tom Brondsted, « The Linguistic Components of the REWARD Dialogue Creation Environment and Run Time System », IEEE 4th Workshop Interactive Voice Technology for Telecommunications Applications (IVTTA'98), Turin, Italie, p. 71-74, 1998.
34. Zouaghi, A., Zrigui, M., Antoniadis, G.: Compréhension Automatique de la Parole Arabe Spontanée. Traitement Automatique des Langues (TAL 2008) 49(1), 141–166, 2008
35. Zouaghi, A., Zrigui, M., Antoniadis, G., Présentation d'un modèle numérique pour la compréhension de la parole arabe spontanée, 2007
36. Zouaghi, A., Zrigui, M., Considération du contexte pertinent pour améliorer les performances d'un étiqueteur sémantique de la parole arabe spontanée, RJC, 2005
37. http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/

Spoken Tunisian Arabic Corpus “STAC”: Transcription and Annotation

Inès Zribi[1], Mariem Ellouze[1], Lamia Hadrich Belguith[1], and Philippe Blache[2]

[1] ANLP Research Group, MIRACL Lab., University of Sfax, Tunisia,
ineszribi@gmail.com, mariem.ellouze@planet.tn, l.belguith@fsegs.rnu.tn
[2] Aix-Marseille Université & CNRS LPL, 13100, Aix-en-Provence, France.
philippe.blache@lpl-aix.fr

Abstract. Corpora are considered as an important resource for natural language processing (NLP). Currently, the Dialectal Arabic corpora are somewhat limited, particularly in the case of the Tunisian Arabic. In recent years, since the events of the revolution, the increasing presence of spoken Tunisian Arabic in interviews, news and debate programs, the increasing use of language technologies for many spoken languages (e.g., Siri) [6], and the need for works on speech technologies requires a huge amount of well-designed Tunisian spoken corpora.
This paper presents the “STAC” corpus (Spoken Tunisian Arabic Corpus) of spontaneous Tunisian Arabic speech. We present our method used for the collection and the transcription of this corpus. Then, we detail the different stages done to enrich the corpus with necessary linguistic and speech annotations that makes it more useful for many NLP applications.

Keywords: Tunisian Arabic, spoken language, corpus transcription, annotation

1 Introduction

The colloquial Arabic or Dialectal Arabic (DA) is the natural spoken variety used in daily communication of the Arabic World and is not generally written. Indeed, there is no commonly accepted standard for colloquial Arabic writing system [9]. Today, processing Arabic and spoken dialects is a new area of research that is faced with many challenges. On the one hand, the oral tradition of the DA and the absence of orthographic standards give rise to the difficulty of its automatic processing. These characteristics engender the lack of written resources for the colloquial Arabic. On the other hand, a number of dialects with linguistic differences on phonological, morphological, syntactic, and lexical levels increase the challenges of building tools and resources for all the Arabic dialects.

The training and testing of statistical or symbolic systems in Natural Language Processing (NLP) require the availability of annotated corpora. Our aim consists on developing resources and tools for the Spoken Tunisian Arabic (STA); one of the North African dialects. In the recent years, many researchers were interested in building dialectal Arabic corpora. Generally, the majority of these

pp. 123–135; rec. 2015-01-25; acc. 2015-02-27

contributions aim to develop textual DA corpora or to set spoken transcriptions without linguistic enrichments. That is why we have decided, as a first step in our work, to build a dialectal corpus of transcribed speech that treats multiple themes. Indeed, since the events of the revolution, the volume of data in Tunisian Arabic (TA) (oral and written) has increased. The TA has become progressively used and represented in interviews, news and debate programs instead of Modern Standard Arabic (MSA) [6]. To constitute the corpus, we are basically based on the available audio in the web in order to transcribe and annotate them. In this paper, we present our collection and transcription method for STA. We detail the different stages done to enrich the corpus with necessary linguistic and speech annotations.

Section 2 presents an overview of Tunisian Arabic. Section 3 presents the main related works. Section 4 is dedicated for presenting the transcription challenges for a language with oral tradition. We describe, in section 5, the collection and the transcription of the TA. Finally, we present, in section 6, the different types of annotation for enriching our corpus.

2 Tunisian Arabic

Tunisian Arabic (TA) is a dialect of the North African (i.e., the Maghreb) dialects spoken in Tunisia [29]. It is considered as a low variety given that it is neither codified nor standardized even though it is the mother tongue and the variety spoken by all the population in daily usage [24]. Approximately eleven millions people speak at least one of the many regional varieties of TA [29].

The linguistic situation in Tunisia is “poly-glossic” where multiple languages (French, Berber, Italian, etc.) and language varieties coexist (MSA and TA) [16]. Indeed, there are many differences as well as similarities points between TA and MSA in different levels [3]. In addition, TA is distinguished by the presence of words from several other languages. The presence of these languages mainly occurred due to historical facts. We find in Tunisia, a numerous examples of several languages; We find a significant number of expressions and words from European languages such as Spanish, French and Italian, Turkish, and even Maltese. In addition, TA contains several words from the vocabulary of Berber language [29]. Likewise, Tunisian people code switch easily and frequently between MSA, TA and the French language in a conversation [30]. This phenomenon allows the introduction of new words (nouns and verbs) derived from foreign languages.

3 Related Works

Today, processing Arabic and spoken dialects is a new area. There are many contributions aiming to develop textual DA corpora ([1], [15], [20], [25], [28], etc.). But, the resources created for Arabic dialects are still in its infancy. The lack of DA corpus is due to the absence of written standards and the lack of

[3] For more details see [29] and [30].

written material for Arabic dialects. We present in this section some works done for building dialectal Arabic corpora.

At present, the major standard Arabic dialects corpora are available through the Linguistic Data Consortium (LDC) by the DARPA EARS program [17] for developing robust speech recognition technology. The LDC provides conversational and broadcast speech with their transcripts. The Levantine Arabic is the object of the LDCs Fisher Levantine Arabic project in which more than 9,400 speakers of the Northern, Southern and Bedwi dialects of Levantine Arabic were participated in collecting 2,000 telephone calls [17]. Furthermore, the data set contains approximately 250 hours of telephone conversations. Each call is up to ten minutes in duration and subjects speak to each other about assigned topics. Moreover, another Arabic colloquial corpus called CALLHOME Egyptian Arabic Speech was dedicated to the Egyptian Arabic ([12], [8]). Indeed, the data set consists of 120 telephone conversations between native speakers of Egyptian dialect. All calls lasted up to 30 minutes. In fact, Egyptian Arabic corpus contains both dialectal and MSA words forms.

As well, the Saudi Arabic dialect was represented by the Saudi Accented Arabic Voice Bank which is very rich in terms of its speech sound content and speaker diversity within the Saudi Arabia [2]. The duration of the total recorded speech is 96 hours distributed among 60,947 audio files. Indeed, the corpus was externally validated and used by IBM Egypt Branch to train their speech recognition engine. English-Iraqi corpus is another Arabic corpora mentioned in [23] and which consists of 40 hours of transcribed speech audio from DARPA’s Transtac program. Furthermore, Almeman et al. [3] have building a multi dialect Arabic speech parallel corpus. This corpus is designed to include three Arabic dialects which are the Egyptian dialect, the Gulf dialect, the Levantine dialects and the MSA. The created corpus is limited to a specific domain which is the travel and the tourism. Their corpus is composed of parallel prompts written for the four main varieties of the Arabic language which involved 32 speech hours. It is composed of written MSA prompts translated to dialects and then recorded [3]. To our knowledge, there is no conventional standard used for transcribing dialects.

In a like manner, Masmoudi et al. [18] and Graja et al. [10] have created speech corpora for STA. To our knowledge, they are the only researchers who created a spoken transcribed corpus for TA. The works done consists on building a corpus for limited domain which gathers a set of dialogues between the staff of the National Company of the Tunisian Railways and the customers who seek information about train schedules, fares, reservation, etc. The sizes of the two corpora are respectively 20 hours and 8 hours of transcribed speech.

4 Transcription Challenges for a Language with Oral Tradition

The transcription task is the operation which consists of replacing each phoneme and sound in the signal audio to a grapheme language of a writing system [19]. Transcription itself is not a trivial task. It requires a series of operations: choice of mode transcription, transcription conventions, segmentation, translation, etc. [19]. Transcribing spoken languages that don't have orthographies rules face to many difficulties. The transcriber is faced to two types of problems. The first problems are related to the nature of the oral speech. This kind of problems is shared between all spoken languages. The second type of the problems is associated to the nature of the treated language which is mainly spoken without written tradition. Indeed, any transcriber is faced to many difficulties caused by the speech perception [19]. The bad quality of the signal or the record, the noisy environment, the presence of too many speakers, the overlaps, etc. are the principal causes of the bad listening of the speech and consequently the presence of the errors in the transcripts. Similarly, many transcribers do not always listen to the same sequence of words as others and altogether they may not hear a word or a sequence of words. The wrong listening prevents the transcriber to write faithfully the realized signal. Despite, the transcriber tries involuntarily to correct partial or incorrect words. He always essays to make sense with the perceived elements. These problem increases when the speech is spontaneous dialogue where speakers intersect speech, hesitate, stutter, etc. So, it is difficult to discern and to identify what whom and when is actually spoken.

On the other hand, TA like other Arabic dialects has no standard orthography. Despite, the TA is a variant of the Arabic language. The relationship between these varieties of Arabic does not prevent the differences between them. It is characterized by a rich lexicon which is a collection of words from Latin, Turkish, Berber, and Maltese origin. The massive borrowing and integration to TA has caused the presence of new phonemes that does not belong to the Arabic language. These particularities of the TA make the transcriber unable to write some words even some expressions. Owing to the absence of orthographic rules, the transcription of TA could be with different characters: Arabic characters, Latin characters, alphanumeric (SMS language), etc. This issue presents another challenge for transcribing TA.

In conclusion, most of the difficulties of oral arise in the preliminary phase for the transcription. There is a competition between different alphabets and therefore several orthographic traditions, and transcription tools, methods, and conventions to adopt [19].

5 Spoken Tunisian Arabic Corpus “STAC”

The transcribing process consists of two basic steps. The first one is providing voice data in order to be transcribed later. The second step consists in its transcription following transcription guidelines. More details about those two steps are described in the following sections.

5.1 Data Collection and Description

The first step in our method for corpus creation is the collection of spoken data. The choice of the speech data content and type is a very important step and could be the key of further use of our corpus. We choose to provide both manually recorded speech audio (part 2) and audio files downloaded from the web (part 1) in order to improve the reuse of our corpus in new research tendency.

Indeed, there are a lot of free resources available on the web. To facilitate the task of creating a corpus for the STA, we are based on the approach "download" and "save" proposed by [27] for searching and downloading audio files who's the speakers express with Tunisian dialects. Following this approach, we recorded 3 hours and 28 minutes of TA speech from different TV channels and radio stations (*Mosaique radio*, *Tunisian national TV*, *Ettounsiya TV* and *Sfax radio*). It presents a first part of our corpus. These streaming are generally radio and television talk shows, debates, and interactive programs where the general public is invited to participate in discussion by telephone. Having a good amount of spoken recordings is fundamental in the design of the corpus. Also, a high sound quality is required and will be useful for other future processing for example in voice recognition system. To keep the good quality of our corpus, we saved only the files in which the speakers can intervene only on one subject simultaneously. Similarly, we choose records with a good sound quality. Sometimes, the quality sound of the recording can vary considerably over time. So, we filter noisy sequences (music or other non-transcribed noise) that last more than one second. The size of the audio files can vary from several tens of seconds to several minutes. We take care that all records contain more spontaneous speech and the percentage of the dialectal content is very higher than MSA or French content.

The second part of our corpus is about 30 minutes taken from the corpus TuDiCoI (Tunisian Dialect Corpus interlocutor) [10] which is a corpus of spoken dialogue in TA and it is obtained from a railway information service. It gathers a set of conversations recorded in the railway station between the staff and customers who request information about the departure time, price, booking, etc. [10]. We have redone the transcription of this part following our convention guidelines.

Including different themes and speakers of TA make our corpus more generic as possible. It contains spontaneous speech, less spontaneous speech and sometimes prepared speech. In addition, the relatively big number of speakers (about 70 speakers) in our corpus speaking each one with its own style make our corpus a representative sample of TA. We provide both individual and multiple speakers in our collection to identify different aspects of conversational speech. Also, the radio and television records have a varied content. There are a wide variety of speakers and themes (social, health, religious, political, and others). Providing speech data with a variety of themes will increase the size of the vocabulary in our corpus and will be very useful for further application for example theme classification [5]. Indeed, we defined the following themes list in our data selection: religious, political, health, social, and others. The corpus contains dialect data from different Tunisian regions. The dialect of Tunis (the capital of Tunisia) is the most dominant while it is the dialect used in the Tunisian media. We consider it as the standard dialect of Tunisia because it is understood by all Tunisian people. It presents about 90% of the totality of our corpus. The table 1 presents some statistics about our corpus.

Table 1. Statistic about STAC corpus.

Themes	**Duration**
Social	01:01:35
Health	01:30:46
Religious	00:12:38
Political	00:50:50
Other	01:14:42
Total	**04:50:31**
TA percentage	97.20%
MSA percentage	0.37%
French language percentage	2.43%

There are a few works done for creating corpus for STA ([18] and [10]). To our knowledge, our corpus is the first resource for TA which contains different types of annotation and enrichments. It can be a good resource spoken for TA and can be used for different purposes. Despite, the size is relatively small. The total size of our corpus is about 5 hours. It is growing, since there are still new recordings that are planned to being done. In conclusion, STAC could be a good textual resource for STA. It could be useful for creating tools and other resources for TA.

5.2 Corpus Transcription

The transcription of spoken speech is a symbolic representation of the spoken language. In the literature, the researchers have defined fours types of transcriptions [19]: phonetic, phonologic, morpho-phonologic, and orthographic. First, the phonetic transcription consists on describing as closely as possible the differences between sounds using the phonetic alphabets: IPA (International Phonetic Alphabet) or SAMPA (Speech Assessment Methods Phonetic Alphabet). It presents the pronunciation of the speech. This type of transcription is expensive in term of time because the transcribers are not familiarized with phonetic alphabets. Second, the phonological transcription consists on describing only the distinctive phonological differences. Then, the morpho-phonologic transcription is a combination between phonological and syntactical notation [19]. It is an analysis and decomposition of all speech constituents. This type of transcription is very expensive in time of writing. Finally, orthographic transcription is a transcription method that employs the standard spelling of the target language. It is easier to the transcriber to use the usual alphabets for a language. After studying these types of transcription, we choose the orthographic transcription. Our choice is justified by different reasons. First, according to our objectives which are the tools development and adaptation in the favour of automatic treating STA, the best choice for our objective is to use the orthographic transcription with the Arabic alphabets. Likewise, the orthographic transcription is easier to the transcriber than to use the phonetic alphabets. Second, we create a corpus, which is easy to read by everybody.

To transcribe our corpus, we choose to use Praat tool[4]. Indeed, the choice was related to our research group needs. This tool allows the analysis of speech in phonetics and also supports speech synthesis, including articulator synthesis[5]. Praat can provide an aligned transcription between speech and text and facilitate the labeling and segmentation of the speech for the linguistic issues due to the tires provided by its interface. The acoustic signal of audio content may correspond to speech, music, noise or/and mixtures of them. Furthermore, in the same record, one or more speakers coexist and discuss many topics in the same time. Also, the recording sound quality may vary significantly over time. As a consequence, it is important to define a set of conventions that specify the orthographic transcription of the different elements which exist in the acoustic signal.

We utilised two works done for TA for transcribing our corpus. The first work is for [30] who presented an Orthographic Transcription of TA (OTTA). The second work is for [29] who proposed a conventional orthography for TA (CODA Tun.) which is an extension of the CODA map [11] to the TA.

Arabic Orthographic Transcription. To standardize the orthographic transcription of the TA, we applied the orthographic transcription conventions of TA defined in OTTA and in CODA Tun. The difference between the two conventions is in the phonetic level. Indeed, CODA Tun. is defined in writing TA words without specifying the phonetic differences between MSA and TA in some cases. Contrariwise, OTTA mixes widely between the phonetic and the orthographic transcription. Hence, we created two versions of our corpus using these conventions. Subsequently, the obtained corpus based on two transcription conventions will be useful for the creation of processing tools for TA such as stemmer, morph-syntactic tagger, etc. Also, it is useful for the creation of automatic speech processing systems such as speech synthesis, automatic transcription systems, etc. [30]. Table 2 presents an example of a TA sentence written according the two orthographic conventions.

Enrichment of the Transcription. Textual transcription of the speech is not sufficient for developing tools and applications for processing automatic language. Moreover, the standard orthographic transcription doesn't take into consideration the observed phenomena of speech (elisions, disfluency, liaison, noise, etc.). So, we add some orthographic enrichment for the transcribed speech. We followed the enriched orthographic transcription guidelines described in the OTTA directive. The guideline proposed by [30] is an adaptation of the Enriched Orthographic Transcription (TOE in French) [4] which is elaborated by LPL laboratory to transcribe conversational French corpus. The French convention is extended with some precisions and modifications for the transcription of STA [30]. We present here some specifications that we used while transcribing our corpus.

Our transcription method consists of an orthographic transcription that it specifies the typical phenomena of the oral. Indeed, the transcript should be close

[4] `http://www.fon.hum.uva.nl/praat/`

[5] `http://en.wikipedia.org/wiki/Praat`

Table 2. An example of sentences in TA.

OTTA	وباش نحكيو زادة عالزيادات في أسوام الڨاز ونسألوا عالاءنعكاسات متاعو عالمواطن.
	wbAš nHkyw zAdħ ςAlzyAdAt fy ÂswAm AlGAz wnsÂ lwA ςAlĂnςkAsAt mtAςw ςAlmwATn.[6]
CODA Tun.	وباش نحكيوا زادة عالزيادات في اسوام القاز ونسألوا عالانعكاسات متاعه عالمواطن.
	wbAš nHkywA zAdħ ςlAlzyAdAt fy AswAm AlqAz wns ÂlwA ςAlAnςkAsAt mtAςh ςAlmwATn.
Translation	*We will talk also about the increase in gas prices and question its impact on the citizen.*

to the signal. We try to write the speech of each speaker and the overlaps speech as possible as we can hear. Therefore, we use neither acronyms nor abbreviation in the transcripts. Similarly, we do not correct the atypical accords. STAC corpus is composed of conversational speech. The script of each speaker is presented separately in an individual tier. Silence pauses could be at the beginning, mixed with the transcript, and at the end of a speaker Turn. We isolated pauses (silent and noisy) which have a minimal duration of 200 ms in a speaking turn. We mark them with the hash tag symbol "#". Also, we mark the silences those are lesser than 100 ms with the plus symbol "+".

Code switching is a main characteristic of the STA. The orthographic form of each non TA word should refer to its orthographic rules. We use this annotation [lan:X, text] for non TA words to be easily recognized (e.g. [lan:FR, deux] for French language). Table 3 describes some annotations used in our corpus.

6 Corpus Annotation

The enrichment of the corpus is key part of NLP. Many systems now are easily developed due the availability of annotated corpora for written and spoken data. Many linguistic annotations could be added to a spoken corpus. Indeed, transcriptions of speech do not contain punctuation marks. Texts contain lexical particularities specific to speech; spoken texts are full of disfluencies. Nevertheless, most of NLP tools should consider these specificities in order to perform the proposed task. For this purpose, we enriched our corpus with different types of annotation (morphosyntactic annotation and disfluencies).

6.1 Morpho-syntactic Annotation

The morpho-syntactic annotation consists on marking up a word in a corpus as corresponding to a particular part of speech, based on both its definition, as well as its context. The manual annotation of the corpus is very difficult and

[6] Transliteration is coded with Buckwalter transliteration. For more details about it, see [13].

Table 3. Some annotations used in STAC corpus.

Transcribed event	**Notation**	**Representation**	**Examples**
Proper name	Between two hooks	[ortho[7], type]	[صفاقس، مك] [*Sfax, place*]
Elisions	The characters related to the omitted phonemes are written between parentheses	otho(c)	ب(ا)ش *not*
Non-linguistic noises and Inaudible sequence	Star	*	* مشى * *he walks*
Laughers	Ampersand	&	&
Tunisian dialect liaison [8]	Between equal signs	otho = letter = ortho	أربعطاش=ن=ألف *fourteen*
Reported speech	Between paragraph symbols	\sequence\	قلت لك\اسكت\ *I told to you* \shut up\
Specific pronunciation	Between square brackets	[ortho, buck[9]]	[جزار, zazza:r] *butcher*
Speech while laughing	Between double ampersands	&&ortho&&	باهي&&*ok ok*&&
Title	Between quotation marks	“ortho”	“عندي ما نق(و)ل لك” *“I have something to say”*
Truncated word	Final dash	ortho-	عس-

expensive in time. Eventually, the development of NLP tools for TA is still in its infancy. We present in this section our method for morpho-syntactic annotation of TA.

The main idea of our method is to disambiguate the output of the morphological analyzer developed for TA, $Al-Khalil$ [31]. Before presenting our method, we should define the tags set that we used for annotating our corpus. Usually, Al-Khalil TA version returns a list of analysis for a given word with different information (gender, prefix, suffix, number, person, voice, POS, etc). For our task, we keep all these morphological characteristics. We add four grammatical categories (i.e. part of speech) which are related to STA. We define NE, FW,

[7] “Ortho” is the orthographic transcription.

[8] The following rules are applicable when we used OTTA transcription guidelines.

[9] “buck” is the transliteration of Buckwalter [13].

FP, and ONOM for respectively named entity, foreign word, filled pause, and onomatopoeia.

The annotation process starts by extracting speaker's text. STAC corpus incorporates the transcription of many conversations between at least two speakers. Hence, speech text for each speaker is divided into many speech turns. We gather the speech turns for each speaker in a unique text. We, then, segment it manually in utterances. We consider an utterance a semantically meaningful unit. The automatic identification of utterances boundaries is considered as a future work. Speech text includes many annotations; some of them are very useful in the morpho-syntactic annotation process. Some of others such as noise and music are removed. Furthermore, all foreign words, filled pauses, onomatopoeia, and also named entities existing in the corpus are not analysed by $Al-Khalil$ [31]. We tagged them respectively as FW, FP, ONOM and NE.

To annotate our corpus, we utilized an iterative procedure to semi-automatic tagging the unannotated data. The iterative procedure starts by dividing our corpus to 10 folders according to the number of sentences. We begin with a morphological analysis of the first folder of the corpus. Eventually, the analyzer gives to each word a set of analysis. We choose the correct analysis according to the position of the word in the utterance. When the analyzer fails in giving an analysis for a word, we determine for this word a set of morphological features. To decrease the number of unrecognized words, we remove all diacritics from the words when analyzing and we enhance also the Al-Khalil lexicon. We train a first version of our POS tagger with the first part of the corpus completely annotated by hand. We applied a multi-class classifier using a rules-based classifier (Ripper) as the main classifier. The generated rules are used for choosing the correct analysis for each word. We use the result tagger for annotating the second folder of the corpus. We manually corrected the output of the tagger, and added the corrected part to the training corpus. Then, we iterate this process over the different parts of the corpus. At the end of this process, we obtain a larger manually corrected corpus (i.e. for each word, all the morphological analysis are kept and only the correct analysis in a given context is marked). The corpus consists of 42,388 words that are transcribed. It is composed of 2,252 verbs, 1,457 nouns, and 458 adjectives. The proposed method achieved an F-measure score of 87%.

6.2 Disfluencies Annotation

Spoken corpus annotation should not be limited to the usual annotations of the written language. Indeed, a speech transcription is a new form of texts with specificities which constitute a practical issue for automatic analysis of spoken texts [7]. So, spoken corpus annotation must take into consideration these specificities, more precisely, the “disfluencies”. Indeed, disfluencies are defined as a phenomenon occurring frequently throughout spontaneous speech, and consist of the interruption of the normal course of speech [14]. In fact, there are different types of disfluencies [22]: filled pauses, repetition of words or word sequences, immediate self-corrections, word fragments. Generally, disfluencies could be combined simultaneously with the association of at least two of phenomena mentioned above. The analysis of disfluencies realized by Shriberg [26] showed that the disfluent sequence can be divided into three regions: the reparandum (word truncation or phrase truncation), break point (filled pauses, silent pauses, etc.) and the repair. Based on Shriberg analysis, Pallaud et al. [21] have defined an annotation schema which reflects the proposed structure of disfluencies. The guideline of annotation is developed in order to annotate disfluencies using Praat. Given the specificities of the TA, especially the code switching, we noticed the presence of some cases of repetition code switching. So, we added some annotations that describe this type of repetition which is specific for the TA. Figure 1 presents an annotated example extracted from our corpus.

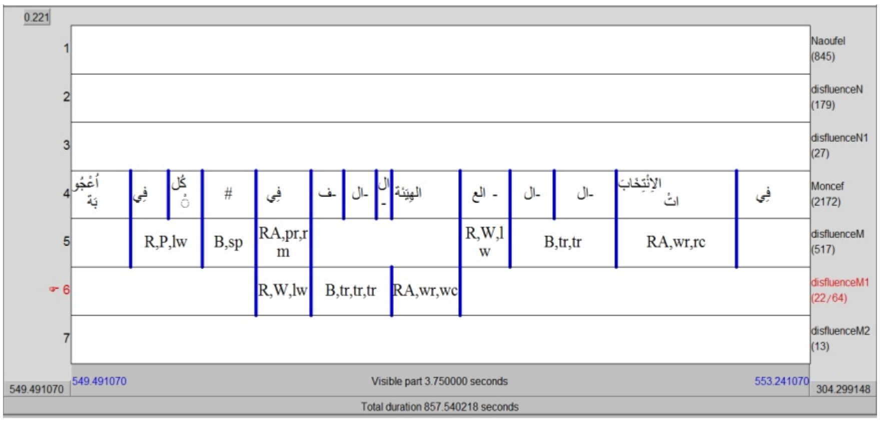


Fig. 1. An example of disfluencies annotation

7 Conclusion and Future Works.

The creation of an annotated corpus presents a challenge for NLP application. Indeed, the Tunisian Arabic is a language with oral tradition without orthographic rules. This issue makes the creation and the annotation of a corpus for this language more difficult. In this context, we presented our effort for the creation of an annotated corpus for spoken Tunisian Arabic. We presented, firstly, the speech data collection and its transcription according to our orthographic transcription guidelines. Then, we describe its enrichment with morpho-syntactic and disfluencies annotations.

Our corpus consists of 5 transcribed hours and we plan to extend it (i.e. increasing the size of the corpus and adding other themes and domains) basing on the proposed methods of transcription and annotation in order to make the corpus as representative as possible of the spoken Tunisian Arabic. In addition, we plan to use this corpus for developing tools for processing Tunisian Arabic.

Developing a POS Tagger for spoken Tunisian Arabic is the main focus at the moment. We intend also to develop a tool that allows the automatic detection of disfluencies in order to consider these phenomena in automatic parsing spoken Tunisian Arabic.

References

1. Al-Sabbagh, R., Girju, R.: YADAC: Yet another Dialectal Arabic Corpus. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, May 23-25. pp. 2882–2889 (2012)
2. Alghamdi, M., Alhargan, F., Alkanhal, M., Alkhairy, A., Eldesouki, M., Alenazi, A.: Saudi Accented Arabic Voice Bank. Journal of King Saud University - Computer and Information Sciences, Riyadh, 20, pp. 45–64 (2008)
3. Almeman, K., Lee, M., Almiman, A.A.: Multi dialect Arabic speech parallel corpora. In: First International Conference on Communications, Signal Processing, and their Applications (ICCSPA). pp. 1–6 (2013)
4. Bigi, B., Péri, P., Bertrand, R.: Orthographic Transcription: which enrichment is required for phonetization? In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, May 23-25. pp. 1756–1763 (2012)
5. Bischoff, K., Firan, C.S., Paiu, R., Nejdl, W., Laurier, C., Sordo, M.: Music Mood and Theme Classification - a Hybrid Approach. In: International Society for Music Information Retrieval (ISMIR 2009). pp. 657–662 (2009)
6. Boujelbane, R., Khemakhem, M.E., Belguith Hadrich, L.: Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora. In: Sixth International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan, October 14-18. pp. 419–428 (2013)
7. Dister, A., Constant, M., Prunelle, G.: Normalizing speech transcriptions for Natural Language Processing. In: 3rd International Conference on Spoken Communication (GSCP'09), Naples, Italy, Feb. pp. 507–520 (2009)
8. Duh, K., Kirchhoff, K.: Lexicon Acquisition for Dialectal Arabic Using Transductive Learning. In: Proceedings of EMNLP'06, Sydney, Australia, July 22-23. pp. 399–407 (2006)
9. Elmahdy, M., Gruhn, R., Minker, W., Abdennadher, S.: Modern Standard Arabic Based Multilingual Approach for Dialectal Arabic Speech Recognition. In: The Eighth International Symposium on Natural Language Processing (SNLP), Bangkok, Thailand (2009)
10. Graja, M., Jaoua, M., Belguith Hadrich, L.: Discriminative Framework for Spoken Tunisian Dialect Understanding. In: Proceedings of Statistical Language and Speech Processing - First International Conference, (SLSP 2013), Tarragona, Spain, July 29-31. LNCS, vol. 7978, pp. 102–110 (2013)
11. Habash, N., Diab, M., Rambow, O.: Conventional Orthography for Dialectal Arabic. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, May 23-25. pp. 711–718 (2012)
12. Habash, N., Eskander, R., Hawwari, A.: A Morphological Analyzer for Egyptian Arabic. In: Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology, Montréal, Canada. pp. 1–9 (2012)
13. Habash, N., Soudi, A., Buckwalter, T.: On Arabic Transliteration. In: Arabic Computational Morphology: Knowledge-based and Empirical Methods (2007)

14. Heeman, P., Allen, J.: Detecting and correcting speech repairs. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 295–302 (1994)
15. Jarrar, M., Habash, N., Akra, D., Zalmout, N.: Building a Corpus for Palestinian Arabic: a Preliminary Study. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing, Doha, Qatar, October 25. pp. 18–27 (2014)
16. Lawson, S., Sachdev, I.: Codeswitching in Tunisia: Attitudinal and behavioural dimensions. Journal of Pragmatics 32(9), pp. 1343–1361 (2000)
17. Maamouri, M., Buckwalter, T., Cieri, C.: Dialectal Arabic Telephone Speech Corpus : Principles, Tool design, and Transcription Conventions. In: NEMLAR International Conference on Arabic Language Resources and Tools, Cairo, Egypt (2004)
18. Masmoudi, A., Estève, Y., Khemakhem, M.E., Bougares, F., Belguith Hadrich, L.: Phonetic tool for the Tunisian Arabic. In: SLTU'2014, Saint-Petersburg, Russia (2014)
19. Moukrim, S.: Morphosyntaxe et sémantique du "présent" Une étude contrastive à partir de corpus oraux Arabe marocain, berbère tamazight et français (ESLO/LCO). Thesis, Université d'Orléans, December (2010)
20. Mubarak, H., Darwish, K.: Using Twitter to Collect a Multi-Dialectal Corpus of Arabic. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), Doha, Qatar, October. pp. 1–7 (2014)
21. Pallaud, B., Blache, P., Bertrand, R.: Codage des annotations de disfluences dans les corpus du CID. pp. 1–5
22. PIU, M., Bove, R.: Annotation des disfluences dans les corpus oraux. In: RÉCITAL 2007. pp. 5–8. Toulouse, France (2007)
23. Precoda, K., Zheng, J., Vergyri, D., Franco, H., Richey, C., Kathol, A., Kajarekar, S.S.: Iraqcomm: a next generation translation system. In: INTERSPEECH 2007, Antwerp, Belgium, August 27-31. pp. 2841–2844 (2007)
24. Saidi, D.: Typology of Motion Event in Tunisian Arabic. In: LingO. pp. 196–203 (2007)
25. Salama, A., Bouamor, H., Mohit, B., Oflazer, K.: YouDACC: the Youtube Dialectal Arabic Comment Corpus. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland (2014)
26. Shriberg, E.E.: Preliminaries to a Theory of Speech Disfluencies. Tech. rep. (1994)
27. Waibel, A., Schultz, T., Vogel, S., Fugen, C., Honal, M., Kolss, M., Reichert, J., Stuker, S.: Towards language portability in statistical speech translation. In: Proceedings of ICASSP'04, May. vol. 3, pp. iii–765–8 vol.3 (2004)
28. Younes, J., Souissi, E.: A quantitative view of Tunisian dialect electronic writing. In: 5th International Conference on Arabic Language Processing, Oujda, Morocco (2014)
29. Zribi, I., Boujelbane, R., Masmoudi, A., Ellouze, M., Belguith Hadrich, L., Habash, N.: A Conventional Orthography for Tunisian Arabic. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'2014), Reykjavik, Iceland, May 26-31. pp. 2355–2361. ELRA (2014)
30. Zribi, I., Graja, M., Khmekhem, M.E., Jaoua, M., Belguith Hadrich, L.: Orthographic Transcription for Spoken Tunisian Arabic. In: 14th International Conference CICLing 2013, Proceedings, Part I, Samos, Greece, March 24-30. LNCS, vol. 7816, pp. 153–163. Springer (2013)
31. Zribi, I., Khemakhem, M.E., Belguith Hadrich, L.: Morphological Analysis of Tunisian Dialect. In: Sixth International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan, October 14-18. pp. 992–996 (2013)

Amazighe Verbal Inflectional Morphology: A New Approach for Analysis and Generation

Fatima Zahra Nejme[1], Siham Boulaknadel[2] and Driss Aboutajdine[1]

[1] LRIT, Unité Associée au CNRST (URAC 29), Université Mohamed V-Agdal, Rabat, Morocco
[2] IRCAM, Avenue Allal El Fassi, Madinat Al Irfane, Rabat-Instituts, Morocco

fatimazahra.nejme@gmail.com, aboutaj@fsr.ac.ma
boulaknadel@ircam.ma

Abstract. Amazighe inflectional morphology poses special challenges to Natural Language Processing (NLP) systems. Its rich morphology and the highly complex word formation process of roots and patterns make NLP tools for Amazighe very challenging. In this paper we present an approach for inflectional morphological analysis and generation for Amazighe verbs. The main motivation for this work is to obtain a linguistically motivated tool based on the concept of patterns and allows, from a verbal entries, to predict the inflectional forms. In this context, the present paper sheds light on three axes. The first axis deals with the levels of the new approach development. The second axis demonstrates the implementation, using finite state transducers, of the extracted rules. The fourth axis discusses experimental evaluations conducted to assess the performance of the analyzer. Our analyzer exploits the efficiency and flexibility offered by finite state machines in modeling while using the NooJ Finite State tools.

Keywords: Amazighe Language, Natural Language Processing, Finite state methods, NooJ, Inflectional Morphology, Verbal Morphology.

1 Introduction

One of the core enabling technologies required in NLP applications is a morphological analyzer. Morphological component is an important language resource for any language technology. It presents a fundamental input to higher levels of linguistic analysis such as syntactic parsing.

Most of the newly investigated languages in NLP are resource-scarce. Amazighe is one of the endangered languages of West Africa. However, the creation of a new governmental institution, namely IRCAM[1] (Royal Institute for Amazighe Culture), has made it possible for the Amazighe language and culture to reclaim their rightful place in many domains and to get its official status. These changes have strengthened the possi-

[1] Institution responsible for the preservation of heritage and the promotion of the Moroccan Amazighe culture and its development (see http://www.ircam.ma/).

bility of promoting the Amazighe language and enabling it to be introduced in the public domain including administration, media and also in the educational[2] system in collaboration with ministries.

Nevertheless, this language, and like most of the languages which have only recently started being investigated for the NLP, still suffers from the scarcity of language processing tools and resources.

Therefore, a set of scientific and linguistic research are undertaken to remedy to the current situation. These researches are divided on two categories: (1) computational resources which include the optical character recognition (OCR) (Amrouch et al., 2010;) , Amazighe corpora (Boulaknadel and Ataa Allah, 2011, Outahajala et al., 2014), and (2) NLP tools which have been limited and carried out on light stemmer (Ataa Allah and Boulaknadel, 2010a), search engine (Ataa Allah and Boulaknadel, 2010b), concordancer (Boulaknadel and Ataa Allah, 2010c), verb Conjugator (Ataa Allah and Boulaknadel, 2014) and morphological analyzer (Nejme et al., 2012a; Nejme et al., 2012b; Nejme et al., 2012c; Nejme et al., 2013a; Nejme et al., 2013b; Nejme et al., 2013c).

Given that the morphological analyzer is regarded as the first in a series of text processing components, this paper presents the continuation of our previous efforts which are designed for the purpose of developing morphological analyzer of Amazighe nouns and particles (Nejme et al., 2013a; Nejme et al., 2013b; Nejme et al., 2013c; Nejme et al., 2012a; Nejme et al., 2012b; Nejme et al., 2012c).

"Amazighe Verbal Inflectional Morphology: a New Approach for Analysis and Generation" is a part of the whole analyzer for Amazighe. It investigates the processing of the verbal morphology using finite state technology within the linguistic developmental environment NooJ. This technology is computationally very efficient for natural language processing.

The remainder of this paper is organized as follows: in Section 2 we provide some features of Moroccan Amazighe language. In Section 3, we present our inflectional system of Amazighe verbs. In Section 4 we give the evaluation results of our experiments while in the last Section we draw some conclusions and highlight our plans for future work.

2 The Amazighe Language

2.1 Historical background

The Amazighe language also known as Berber or Tamazight (□□□□□□□□ [tamaziɣt]), is belongs to the African branch of the Afro-Asiatic language family, also referred to Hamito-Semitic in the literature (Greenberg, 1966; Ouakrim, 1995). Geographically speaking, it covers the Northern part of Africa which extends from the Red Sea to the Canary Isles and from Niger in the Sahara to the Mediterranean Sea.

In linguistic terms, the language is characterized by the proliferation of dialects due to historical, geographical and sociolinguistic factors. In Morocco, one may distinguish three major dialects: Tarifit in the North, Tamazight in Central Morocco and South-

[2] It has become common practice to find Amazighe taught in various Moroccan schools as a subject.

East, and Tachelhite in the South-West and the High Atlas. Since the ancient time, it is the mother tongue of approximately half of the population. However for many decades, it was, until 1994, only oral exclusively reserved for family and informal domains (Boukouss, 1995). While by the creation of the Royal Institute of Amazighe Culture (IRCAM) in 2001 and the constitution update of July 2011, the status of Amazighe has progressively changed.

The Amazigh language has its own script called Tifinaghe that was adapted by the Royal Institute of the Amazighe Culture (IRCAM) in 2003, to provide an adequate and usable standard alphabetic system called Tifinaghe-IRCAM. This system contains:

- 27 consonants including: the labials (□, □, □), dentals (□, □, □, □, □, □, □, □), the alveolars (□, □, □, □), the palatals (□, □), the velar (□, □), the labiovelars (□□, □□), the uvulars (□, □, □), the pharyngeals (□, □) and the laryngeal (□);
- 2 semi-consonants: □ and □;
- 4 vowels: three full vowels □, □, □ and neutral vowel (or schwa) □ which has a rather special status in Amazighe phonology.

Today, the current situation of the Amazighe language is at a pivotal point. It holds official status beside Arabic. Its morphology as lexical standardization process is still underway. At present, it represents the model taught in must schools and used on media and official papers.

2.2 Amazighe Morphology: A Brief Overview

Amazighe morphology is well-known for being rich and complex in terms of its high inflections and derivations involving infixation, prefixation and suffixation (Boukhris et al., 2008). Amazighe morphology has a multi-tiered structure and applies non-concatenative morphotactics. Words in Amazighe are originally formed through the amalgamation of roots and patterns, as shown in Table 1. A root is a sequence of one or many consonants and the pattern is a template of vowels (V) with slots into which the consonants (C) of the root are inserted. This process of insertion is called interdigitation. The resulting lemmas then pass through a series of affixations (to express morpho-syntactic features) and/or clitic attachments (as personal marker: □□□□□ [tinid] "you say" → the 2nd singular personal marker in the indicative mood □--□ [t--d] affixed to the verb □□□ [ini] "she said") until they finally appear as surface forms.

Table 1. Example of words made following a pattern.

Root	□□ [gl]					
Category	N	N	N	V	V	V
Pattern	uCC	aCCu	aCC'C'a	aCC	CCCaCC	CCCaCaC
Radical	□□□ [ugl - tooth]	□□□□ [aglu-gizzard]	□□□□□ [aglla-flank]	□□□ [agl - suspend]	□□□□□□ [ttwagl - be suspended]	□□□□□□□ [ttyagal- be suspended]

The Amazighe morphology covers three main lexical categories, which are noun, verb, and particles[3] (Boukhris et al., 2008). The focus of this work is practically on verbal morphology.

Amazighe Verbal Morphology

Practically speaking, verbs are the base of the Amazighe morphology because (1) it represents a wide morphological class which is remarkably rich and (2) as others can be derived from them. It is classified according to the number of consonants of their lexical root: there are Monoliteral verbs, Biliteral, Triliteral ones etc. The verb occurs in two forms: basic and derived one. The basic form (radical) is formed through an amalgamation of a root and a pattern (Root: □□ [gl], Pattern: aCC, Radical: □□□ [agl] "suspend"). While, the derived one is obtained by the combination of a basic verb with one of the following derivational morphemes: □/□□ [s/ss] indicates the factitive form, □□ [tt] marks the passive form and □/□□ [m/mm] designates the reciprocal.

The verb, whether basic or derived, inflects in four aspects namely: aorist, perfective, negative perfective and imperfective, that is marked with vocalic alternations, prefixation or consonant gemination/degimination. Moreover, it displays three moods (indicative, imperative and participial), where in each mood the same personal markers are used (cf. Table 3). The indicative and the participial moods are based on the four aspects, while the imperative mood has two forms simple and intensive that are based respectively on the aorist and the imperfective aspects (Boukhris et al., 2008).

In general, the verbal form of an Amazighe verb can be inflected as shown in Table 2 and described using the template of Figure 1.

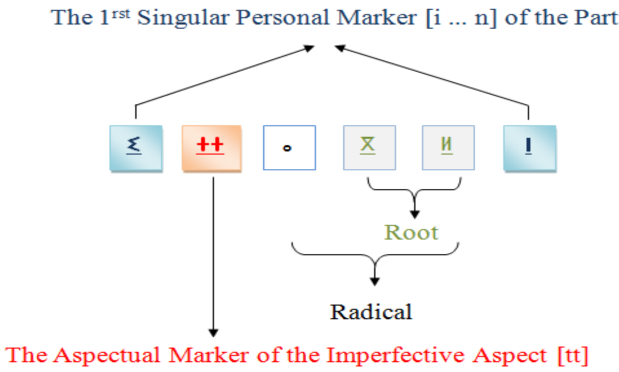

The 1rst Singular Personal Marker [i ... n] of the Participial Mood
ⵉ
ⵜⵜ
ⴰ
ⴳ
ⵍ
ⵏ
Root
Radical
The Aspectual Marker of the Imperfective Aspect [tt]

Fig. 1. Example of template describing an Amazighe verbal form for the first singular person.

[3] That includes all other morpho-syntactic categories other than noun and verb.

Table 2. The inflectional forms of the verb □□□ [agl] “suspend” in the three moods for the 2nd person masculine plural.

	Aspects					
Moods	**Aorist**	**Imperfective**	**Perfective**	**Negative Perfective**	**Simple**	**Intensive**
Indicative	□□□□ [aglɣ]	□□□□□□ [ttaglɣ]	□□□□ [uglɣ]	□□□□□ [ugilɣ]	-	-
Participial	□□□□□ [ya-gln]	□□□□□□□ [ittagln]	□□□□□ [yugln]	□□□□□□ [yugiln]	-	-
Imperative	-	-	-	-	□□□ [agl]	□□□□□ [ttagl]

Table 3. Personal markers for the indicative, imperative and participial moods.

		Indicative mood			Imperative mood		Participial mood
		Masc.	Fem.		Masc.	Fem.	Masc./ Fem.
Singular	1st pers.	...□	...□	2nd pers.	...Ø	...Ø	□…□
	2nd pers.	□…□	□…□				
	3rd pers.	□…	□…				
Plural	1st pers.	□…	□…	2nd pers.	…□□/□/□	…□□□/□□	…□□□
	2nd pers.	□…□	□…□□				
	3rd pers.	…□	…□□				

In the following, we will concentrate on the aspects of verbal inflections – i.e. how inflectional aspects are expressed and generated-.

3 Inflectional Morphology of Amazighe Verbs: Conjugation rules

This section aims to review the formation processes of the verbal aspect: aorist, perfective, negative perfective and imperfective.

3.1 Related Work

Amazighe verbal morphology is already shown in several previous studies. The first exploration refers to the works which tend to concentrate on particular dialects as Ait

Attab dialect (Iazzi, 1991), Imdlawn Tashlhit (Dell and Elmedlaoui, 1985; Dell and Elmedlaoui, 1989) and Ait Ayache one (Ernest, 1971) and which support that the morphological rules that govern the formation of different verb forms take as a basis the lexical entry of the verb that has two types of information: CV template and melodies. However, these studies are insufficient because they deal with a region within a geographical border.

The second one is relies on study of the various conjugation structures inferred of the three major Moroccan Amazighe dialects (El gholb, 2009; Laabdelaoui et al., 2012). El gholb (El gholb, 2009), given the extent of the Amazighe language on a huge geography, has chosen some representative dialects of the three major ones on sporadic basis in order to give an overview of all relevant changes. Based on this result, he has presented a draft in which he adopts the classification by verbal type: monoliteral, bilateral, trilateral, etc. but limited only to the conjugation of simple and underived verb of monoliteral and bilateral types with the structures: /ccv/, /c'c'v[4]/, /vcc/, /vc'c'/, /vcv/, /cvc/, /vc/. Laabdelaoui et al (Laabdelaoui et al., 2012) adopt the class based approach. The verbs are arranged into 31 classes along the aorist/perfective, and the aorist/imperfective conjugation oppositions. In the first 30 classes, independently of the morph phonological alternations, all verbs belonging to a specific class are modeled by the same morphotactic rules to get either the perfective or the imperfective forms, whereas the last class contains a set of 10 verbs that behave differently. Based on these classification criteria, the Amazigh verb and its derived forms do not necessarily belong to the same class, since they may not use the same morphotactic rules to be conjugated.

Class based approach provides a straightforward way of describing a large number of verbs in a compact and generalized way but fails to predict the class for a new verbs (other than those owned by our list and also for the derived forms resulted) or how it forms are morphologically generated. Also, and given the nature of Amazighe morphology, it accounts many classes (31 classes) and, deal with regional varieties, may present different conjugation classes for the same verb.

3.2 Our Approach

Our Approach investigates the mechanism responsible for predicting the conjugation of Amazighe verbs in each aspect. Furthermore, we proposed that the inflection of verbal aspects is based on the pattern. In this context, we have undertaken to develop a set of rules to generalize the inflection model of each pattern.

In line with our goal, and with the aim of the representativeness of the three Moroccan varieties (Tarifit, Tamazight, Tashelhit), we have adopted, as basis of our work, a set of 3676 attested and standardized word lemmas from (Laabdelaoui et al., 2012). This cross-dialectal perspective has several advantages, the main one being that it contributes to a clearer description of the system and allows highlighting the characteristics commonly shared by the different dialects, in order to present the variations that occur.

Starting from this basic list, and in order to simplify the presentation and account for different verbal bases, we choose a classification according to the verbal types (monoliterals, biliterals, etc.), containing a vowel or not and containing geminate radical or not. Then, we extract the rules for each pattern of each type.

[4] We use the "C'" in the pattern presentation when a consonant is reduplicated.

The following diagram (cf. Figure 2) demonstrates the overall architecture of our approach.

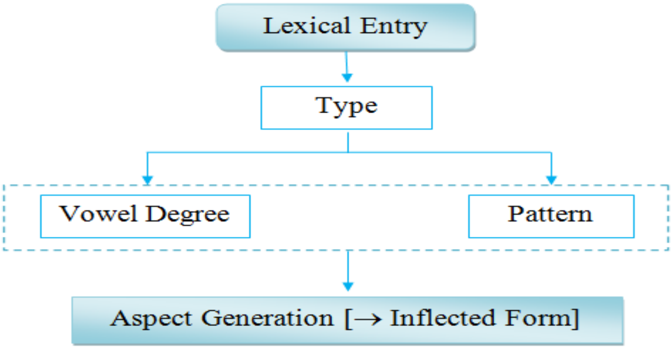


Fig. 2. Verbal inflection architecture.

Our approach is based on hierarchical structures:

- The first phase is to determine, from each entry, the verbal type,
- The second one is to determine the vowel degree (zero vowels or full ones) with the pattern (CV),
- The third one, and based on these two latest, we determine the changes that need to be assign to generate the inflected forms for each aspect.

In order to better illustrate our proposition, we consider as an example the lexical entry □ □ □ [agl] “suspend”, which correspond to the biliteral type with the pattern “aCC” (full vowels). For this template, the inflected forms are generated as follows:

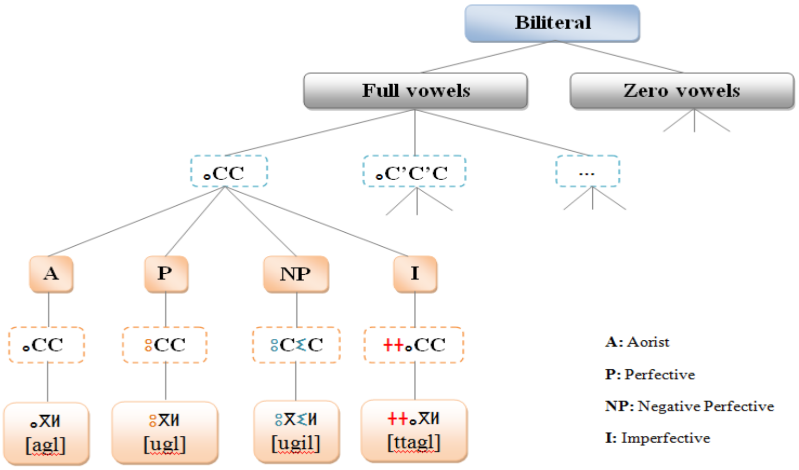


Fig. 3. Inflected forms of the lexical entry □ □ □ [agl]

Based on this classification, a set of paradigms were carefully developed to cover six verbal types and also to present the verbs inflectional exceptions which require a particular study. As a result, we have raised a set of 553 general rules: 329 for regular verbs and 224 for exceptional ones. The following table (cf. Table 4) describes these rules in more details.

Table 4. Description of our rules.

Types	**Pattern**							
	Zero vowels				**Full vowels**			
	Regulars		**Exceptions**		**Regulars**		**Exceptions**	
	Number of entries	**Number of rules**	**Number of entries**	**Number of rules**	**Number of entries**	**Number of rules**	**Number of entries**	**Number of rules**
Monoliterals	5	2	2	2	59	16	9	9
Biliterals	117	5	22	10	570	73	72	44
Triliterals	766	6	64	15	555	88	150	73
Quadriliterals	285	5	63	16	680	90	100	42
Quinquiliterals	21	3	13	5	119	35	10	8
Six literals	1	1	-	-	7	5	-	-
Total	1195	22	164	48	1990	307	341	176

4 The Processing of Verbal Morphology: Implementation and Evaluation

Our main agenda is to develop a highly flexible verbal Amazighe morphological analyzer consisting of two major parts:

(1) Construction of our verb Amazighe lexicon “VAmLex” which stands for “Verbal Amazighe lexicon”,

(2) And the formalization of the inflectional morphology raised by our approach.

To achieve this goal, we use finite state machines within the developmental environment NooJ (Silberztein, 2005; Silberztein, 2006). The use of finite-state tools was extremely attractive, it used to generate and analyze several thousands of words per second. This linguistic platform will be described inside this paper as the tool used for formalization and morphological analysis of Amazighe verbs.

4.1 NooJ: Linguistic Development Framework

NooJ[5], released in 2002 by Max Silberztein (Silberztein, 2005; Silberztein, 2006), is a freeware language-engineering development environment, runs on different operating systems such as Windows, Linux and Mac OSX, and provides a set of tools and methodologies for formalizing and developing a set of Natural Language Processing (NLP) applications. It presents a package of finite state tools that integrates a broad spectrum of computational technology from finite state automata to augmented/recursive transition networks. This package allows constructing, testing and maintaining large-coverage lexical resources, as well as electronic dictionaries, formal, morphological and syntactic grammars, which can be applied to treat texts and large corpora in order to locate morphological, lexicological and syntactic patterns, remove ambiguities, and tag simple and

[5] See http://www.nooj4nlp.net/ for information of NooJ.

compound words. For each of these levels, NooJ provides linguists with one or more formal framework specifically designed to facilitate the description of each phenomenon, as well as parsing, development and debugging tools designed to be as computationally efficient as possible, from Finite-State to Turing machines.

One of the important and useful features of NooJ, regarding morphologically rich languages like Amazighe, is its simple description of morphological and syntactic phenomena, efficient morphological processing, its robustness, and also its ability to analyze texts of several million words in real time. According to all these reasons, we have chosen NooJ as our development platform for building local and morphological grammars that should function as verb morphological analyzer. This component would allow us to process and take advantage of this readily available data.

4.2 Verb Lexicon and Rules Formalization

Given that the linguistic resources required by the morphological analyzer include a lexicon and inflection rules for all paradigms, we started by building a verbal morphological Amazighe lexicon.

Verbal lexicon

Lexical entries were developed from Amazighe Conjugation Manual [□□□□□ □ □□□□□ □ □□□□□□□ - adlis n usfti n tmaziṛt] (Laabdelaoui et al., 2012) and also from the new grammar of Amazighe (Boukhris, 2008). Our main lexicon contains, actually, 3676 entries (3166 regular[6] verbs and 510 irregular[7] ones) represented as a second person, singular, masculine and imperative mood.

Each lexical entry presents the following details: the lemmas, lexical category, type, semantic feature and the translation in French and Arabic languages. Furthermore, each one is linked to its inflection rule invoked by the property "+FLX=" for the inflectional information.

Rules Formalization

The purpose of this section is the verbal formalization. This study presents the implementation of inflectional rules allowing generating from each entry its inflected forms.

Inflectional Rules

Relied on the rules presented in the Amazighe Conjugation Manual (Laabdelaoui et al., 2012) and following our approach we have formalized the verbal inflectional rules. Therefore, we have created, through hand-encoded graphs integrated in the linguistic development platform NooJ, a set of hand-encoded inflectional paradigms covering the exceptional cases. The inflectional descriptions include the mood (indicative, imperative or participial), the gender (masculine or feminine), the number (singular, plural), the aspect (aorist, imperfective, positive perfect and negative one) and the person (first, second or third). By these descriptions we refer to the set of all possible transformations

[6] Verbs with aorist formally identical to the perfective.

[7] Verbs with aorist formally different to the perfective.

which allow us to obtain, from a lexical entry, all inflected forms. On average, there are 82 inflected forms per verb entry and 357805 fully inflected forms in the total.

To give an overview of all these rules, we take as an example the verb □□□ [agl] “suspend”.

□□□,V+Simple+Bilitère+Irreg+Tr+FLX=aC'C'_aff+FR=suspendre+accrocher+être suspendu+AR=علّق +تعلّق

Inflectional grammar is looking for the paradigm named “aC'C'_aff” in order to generate all forms of a headword. Among the 82 inflectional transformations which are described in the inflectional paradigm “aC'C'_aff”, here is one:

□<L2>□<LW>□<S>□/Acc_Négatif+2+m+s

This NooJ paradigm, written in NooJ graphic editors, consists of a number of pairs describing all the possible forms. The first part of this pair describes a change on the word (e.g. <LW>/ - position the cursor (|) at the beginning of the form, <L2>/ - go left with 2 character and <S>/- delete next character) while the second part describes features that the newly made word is given (e.g. / Acc_Négatif+2+m+s– verb is added description that is in negative perfective form (Acc_Négatif), second person (2), masculine (m) and singular (s)). The meaning of the transformation is: (1) to alternate the first vowel □ [a] → □ [u], (2) insert the vowel □ [i] before the last consonant and (3) finally, add the personal markers of the negative imperfective aspect which correspond to the second masculine person. These operations, applied in succession, generate the form: □□□□□□ (tugild– it has not suspended).

4.3 Experiment and evaluation

The performance evaluation of a morphological analyzer has to be observed in terms of its impact on the performance of the applications that use it. Hence, the main goal of this experiment is to prove the flexibility of our approach and to prove that it can satisfy the morphological analysis of the most verbs from the patterns. To do this, at the end of the development phase, we have carried out the evaluation of our inflectional rules against a list of 701 distinct verbs. The list entries, manually constructed, were not used as part of the development of our rules in order to get some feedback and to improve the modeling of inflectional morphology of Amazighe verbs.

After the application of the inflectional rules to the verbs list, we have undertaking a manual analysis of the output to evaluate the performance of our rules. The results for full analysis can be seen in the following table (cf. Table 5).

Table5. Amazighe verbal analyzer evaluation.

Results	**Verbs correctly analyzed**		**Verbs incorrectly analyzed**	
	Number	**%**	**Number**	**%**
Inflectional rules	591	84,30%	110	15,69%

The above results indicate that our verbal system in its current development has so far registered success. Out of the lists respectively of 701 distinct verbs, 84,30% were found to be correctly inflected.

By taking a closer look at the verbs which were not correctly analyzed we could come up with the following conclusions the incorrect analyses are mostly due to: (1) 30% of verbs which patterns are not included in those already treated and (2) to 70% of incorrect inflections. A high rate of this part represents the difference in the Imperfective form (with correct perfective and negative perfective). This difference is due to verbs of some regional varieties of the Amazighe language. But the inflections remain correct for the standard side.

5 Conclusion and future works

We presented a high accuracy morphological analyzer for Amazighe verbs that exploits the regularity in the inflectional paradigms while employing the NooJ Finite State tools for modeling the language in an elegant way. The research results presented above describe the first efforts aimed to investigate the mechanism responsible for predicting the conjugation of Amazighe verbs based on patterns.

The accuracy figures as high in evaluation of our method seem to be appropriate and encouraging. These results allow us to review, correct and complete all our resources in order to improve it.

In order to emphasize more on the usefulness of our approach towards morphological analysis of Amazighe verbs, we plan to add new verbal lemmas and specific tags in order to enlarge the lexicon and to handle the regional varieties. Furthermore, the incorrect forms and the new patterns will be re-examined for further consideration into the morphological system.

References

Amrouch, M., Rachidi, A., El Yassa, M., Mammass, D. 2010. *Handwritten Amazighe Character Recognition Based On Hidden Markov Models*. International Journal on Graphics, Vision and Image Processing. 10(5), pp.11—18.

Ataa Allah, F., Boulaknadel, S. 2010a. Pseudo-racinisation de la langue Amazighe. In Proceeding of Traitement Automatique des Langues Naturelles. Montréal, Canada.

Ataa Allah, F., Jaa, H. 2009. Etiquetage morphosyntaxique: Outil d’assistance dédié à la langue Amazighe. In Proceedings of the 1er Symposium international sur le traitement automatique de la culture Amazighe. Agadir, Morocco, pp. 110- -119.

Ataa Allah, F., and Boulaknadel, S. 2010b. Amazigh Search Engine: Tifinaghe Character Based Approach, Proceeding of International Conference on Information and Knowledge Engineering. Las Vegas, NV, July 2010.

Boulaknadel, S., Ataa Allah, F. 2011. Building a standard Amazighe corpus. In Proceedings of the International Conference on Intelligent Human Computer Interaction. Prague, Tchec.

Bouhamidi, F. 1985. La dérivation verbale dans le parler des Ait Morghad.

Boulaknadel, S., Ataa Allah, F. 2010. Online Amazighe Concordancer. In Proceedings of International Symposium on Image Video Communications and Mobile Networks. Rabat, Maroc.

Boukhris, F., Boumalk, A., Elmoujahid, E. H., and Souifi., H. 2008. La nouvelle Grammaire de l'Amazighe. Rabat: IRCAM.

Boukous, A. 1995. Société, Langues et Cultures au Maroc: Enjeux Symboliques. Casablanca: Najah El Jadida.

Dell, F., Elmedlaoui, M. 1985. *Syllabic Consonants and Syllabification in Imdlawn Tashlhiyt Berber*. Journal ofAfrican Languages and Linguistics 7: 105-130.

Dell, F., Elmedlaoui, M. 1989. Clitic Ordering, Morphology and Phonology in the Verbal Complex of Imdlawn Tashlhiyt Berber, Part I. Langues Orientales Anciennes Philologie et Linguistique 2 : 165-194.

El Gholb, L. 2009. La conjugaison du verbe en amazighe: élément pour une organisation, mémoire de Master 2, Université Ibn Zohr, Agadir.

Es Saady, Y., Rachidi, A., El Yassa, M., Mammas, D. 2010. *Printed Amazighe Character Recognition by a Syntactic Approach using Finite Automata*. International Journal on Graphics, Vision and Image Processing, 10(2), pp.1—8.

Ernest. T. A. 1971. Tamazight verb structure : a generative approach. Volume 2 de African series, Indiana University publications.

Fakir, M., Bouikhalene, B., Moro, K. 2009. Skeletonization methods evaluation for the recognition of printed tifinaghe characters. In Proceedings of the 1er Symposium International sur le Traitement Automatique de la Culture Amazighe. Agadir, Morocco, pp. 33—47.

Greenberg, J. 1966. The Languages of Africa. The Hague.

Iazzi, M. 1991. Morphologie du verbe en tamazight (parler des Aït Attab, Haut Atlas Central), approche prosodique. Thèse de DES, Université Mohamed V, Rabat.

Laabdelaoui, R., Boumalk, A., Iazzi, E. M., Souifi, H., Ansar, K. 2012. Manuel de conjugaison de l'amazighe/ adlis n usfti n tmazivt, Rabat: Publications de l'Institut royal de la culture amazighe.

Naït-Zerrad, K. 1994. Manuel de conjugaison kabyle, Paris, L'Harmattan.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2013a. Analyse Automatique de la Morphologie Nominale Amazighe. Actes de la conférence du Traitement Automatique du Langage Naturel (TALN). Les Sables d'Olonne, France.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2013b. Finite State Morphology for Amazighe Language. In Proceeding of International Conference on Intelligent Text Processing and Computational linguistics (CICLing). Samos, Greece.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2013c. Toward a noun morphological analyser of standard Amazighe. In Proceeding of International Conference on Computer Systems and Applications (AICCSA). Fes, Maroc.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2012a. Toward an amazigh language processing. Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), COLING 2012, Mumbai, December, pages 173 - 180.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2012b. Formalisation de l'Amazighe standard avec NooJ. Actes de la conférence JEP-TALN-RECITAL. Grenoble, France.

Nejme, F., Boulaknadel, S., Aboutajdine, D. 2012c. Vers un dictionnaire électronique de l'Amazighe. Actes de la conférence internationale sur les Technologies d'Information et de Communication pour l'AMazighe 2012c. Rabat, Maroc.

Outahajala, M., Zekouar, L., Rosso, P., Martí, M.A. 2010. Tagging Amazighe with AnCoraPipe. In Proceeding of the Workshop on Language Resources and Human Language Technology for Semitic Languages. Valletta, Malta, pp. 52—56.

Ouakrim, O. 1995. Fonética y fonología del Bereber. Survey at the University of Autònoma de Barcelona.

Sadiqi, F. 1997. Grammaire du berbère, Paris, L'Harmattan.

Silberztein, M. 2005. NooJ : The Lexical Module In NooJ pour le Traitement Automatique des Langues. S. Koeva, D. Maurel, M. Silberztein Eds, MSH Ledoux. Franche-Comté Academic Presses.

Silberztein, M. 2006. NooJ Manual. Download from http://www.nooj4nlp.net.

Development of Amazighe Named Entity Recognition System Using Hybrid Method

Meryem Talha, Siham Boulaknadel, Driss Aboutajdine

LRIT, Associate Unit to CNRST, Faculty of Science, Mohammed V University
Rabat, Morroco
Royal Institut of Amazighe Culture, Allal El Fassi Avenue, Madinat Al Irfane,
Rabat-Instituts, Morroco
CNRST, Angle FAR Avenues and Allal El Fassi, Hay Riad, BP 8027 NU, 10102
Rabat, Morocco
meriem.talha@gmail.com, boulaknadel@ircam.ma, aboutaj@fsr.ac.ma

Abstract. The Named Entity Recognition (NER) is very important task revolving around many natural language processing applications. However, most Named Entity Recognition (NER) systems have been developed using either of two approaches: a rule-based or Machine Learning (ML) based approach, with their effectiveness and weaknesses. In this paper, the problem of Amazighe NER is tackled through using the two approaches together to produce a hybrid system with the aim of enhancing in general performance of NER tasks. The proposed system is able of recognizing 5 different types of named entities (NEs): Person, Location, Organization, Date and Number. It was tested on a corpus of Amazigh reports containing 867 diverse articles. Furthermore, a comparison with the baselines of the system based on the case of using just gazetteers and hand-written heuristics is presented. We also provide the detailed analysis of the results.

Keywords: Amazighe Language, Named Entity Recognition (NER), Hybrid Method, GATE

1 Introduction

Named Entity Recognition (NER) is an important subfield of the broader research area in Information Extraction from textual data, aimed at identifying and associating just some types of atomic elements in a given text to a set of predefined categories such as names of persons, organizations, locations, dates, and quantities, called Named Entities (NE)[1]. It serves as the basis for many other crucial areas such as Information Processing & Management[2], financial documents[3], business information documents[4] and biomedical texts[5], particularly involving information retrieval [6]; semantic annotation[7]; classification; ontology population[8]; opinion mining[9], filtering and summarization[10]; question answering[11]; machine translation[12], browsing and visualization; and human-computer interaction in information systems. The term Named Entity

pp. 151–161; rec. 2015-01-24; acc. 2015-02-27

was first used at the 6th Message Understanding Conference (MUC)[13], where the importance of the semantic identification of persons, organizations and localizations, as well as numerical expressions such as time and quantities was obvious. Although the task is given considerable research attention for so many languages including English, French, Spanish, Chinese, and Japanese, etc.

Named entity recognition research on Amazighe texts is known to be scarce. To the best of our knowledge, [14] present the first study on the topic where a rule based named entity recognition system is proposed and evaluated on an Amazighe corpus which contains 200 Amazighe texts, the system was able to extract 3 different types of NEs including Person, Location, Organization. As a continuation of the previous research work, [15] have presented a system which carries out named entity recognition using a set of heuristic rules and lexical resources, they evaluated their system on a corpus containing 289 texts, that can recognize five NE types including Person, Location, Organization, Expressions of Time, Numbers. Lastly, In [16], authors selected 430 Amazighe texts to work on, and they employed a set of lexical resources and sets of rules as information sources, they obtained remarkable results in the detection of Person, Location, Organization, Expressions of Time and Number entities. In this paper, we present a hybrid named entity recognizer for Amazighe texts.

The remainder of the paper is organized as follows. Section 2 presents a background of Amazighe language features illustrating the challenges posed to NER. In section 3 we discuss the details of our approach including system architecture and the machine learning algorithm used, experimental sets and results obtained are shown in section 4. Finally, we discuss the results and some of our insights in section 5.

2 Amazighe Language Features

The Amazighe language, known as Berber or Tamazight, is a branch of the Afro-Asiatic (Hamito-Semitic) languages [17][18]. In Morocco, this language is divided, according to historical, geographical and sociolinguistic factors, into three main regional varieties: Tarifite in North, Tamazight in Central Morocco and South-East, and Tachelhite in the South-West and the High Atlas. However in 2001, thanks to IRCAM[19] efforts, the Amazigh language has become an institutional language nationally recognized; and in July 2011, it has become an official language besides the classical Arabic.

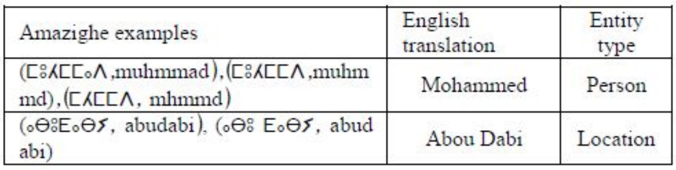
Nowadays, The Tifinaghe-IRCAM graphical system has been adapted in writing Amazighe language for technical, historical and symbolic reasons. It is written from left to right and contains 33 alphabets (27 consonants; 2 semi-consonants and 4 vowels)[20].

2.1 Challenges Undertaken by Amazighe NER

A lot of Named Entity Recognition Systems have been already done thanks to the impulse of MUC conferences. However most of these works have been concentrated on English and other European languages. Yet, named entity recognition research conducted on Amazighe texts is still rare as compared to related research carried out on other languages.

In particular, Applying NLP tasks to Amazighe are very challenging because of its particularities and unique nature. The main features of Amazighe that pose non-trivial challenges for NER task are as follows:

- No Capitalization: The absence of the uppercase / lowercase distinction represents a major obstacle for the Amazighe language. In fact, the NER for some languages such as Indo-European languages is mainly based on the presence of capital letters which is a very useful indicator to identify proper names in major languages using the Latin alphabet. Uppercase letters, however, do not occur, neither at the beginning neither at the initial of Amazighe names.
- Complex Morphological System: It is a fact that the Amazighe language is agglutinative having a rather complex and rich derivational and inflectional morphology. Names can have several inflected and derived forms; a simple elimination of suffixes is not enough to reunite words families. Indeed, affixes can alter the meaning of a word.
  Similarly to other natural languages, Amazighe presents uncertainties in grammatical classes. Actually the same form is suitable for numerous grammatical categories, depending on the context in the sentence. For example, ”illi” (transliterated in a french-style) can be considered as an accomplished positive verb, it means ”there is” or as the name of kinship ”my daughter.”
- Spelling Variants: The Amazighe language has remained essentially an oral language for a long time. Therefore, the Amazighe text does not respect the standard writing convention. Furthermore, Amazighe text contains a large number of transliterated and translated NEs. These translated and transliterated words may be spelled differently and still refer to the same word with the same meaning, producing a many-to-one ambiguity. Fig. 1 shows some examples.



Amazighe examples	English translation	Entity type
(ⵎⵓⵃⵎⵎⴰⴷ ,muhmmad),(ⵎⵓⵃⵎⵎⴷ ,muhmmd),(ⵎⵃⵎⵎⴷ, mhmmd)	Mohammed	Person
(ⴰⴱⵓⴷⴰⴱⵉ, abudabi), (ⴰⴱⵓ ⴷⴰⴱⵉ, abud abi)	Abou Dabi	Location

Fig. 1. Examples of Variations in Amazighe Texts

- Lack of Linguistic Resources: We lead study on the Amazighe language resources and NLP tools (e.g., corpora, gazetteers, POS taggers, etc.). This led us to wrap up that there is a limitation in the number of available Amazighe linguistic resources in comparison with other languages. Many of those available are not relevant for Amazighe NER tasks due to the absence of NEs annotations in the data collection. Amazighe gazetteers are rare as well and limited in size. Therefore, we tend to build our Amazighe linguistic resources in order to train and evaluate Amazighe NER systems.

3 Amazighe NER System Architecture

In this paper, we develop an hybrid architecture that is normally better than the rule-based or machine-learning systems individually. Figure 2 illustrates the architecture of the hybrid NER system for Amazighe.

The system consists of two modes: rule-based and ML-based Amazighe NER modes. The processing goes through three main phases: 1) The rule-based NER phase, 2) The feature selection and extraction, and 3) the ML-based NER phase.

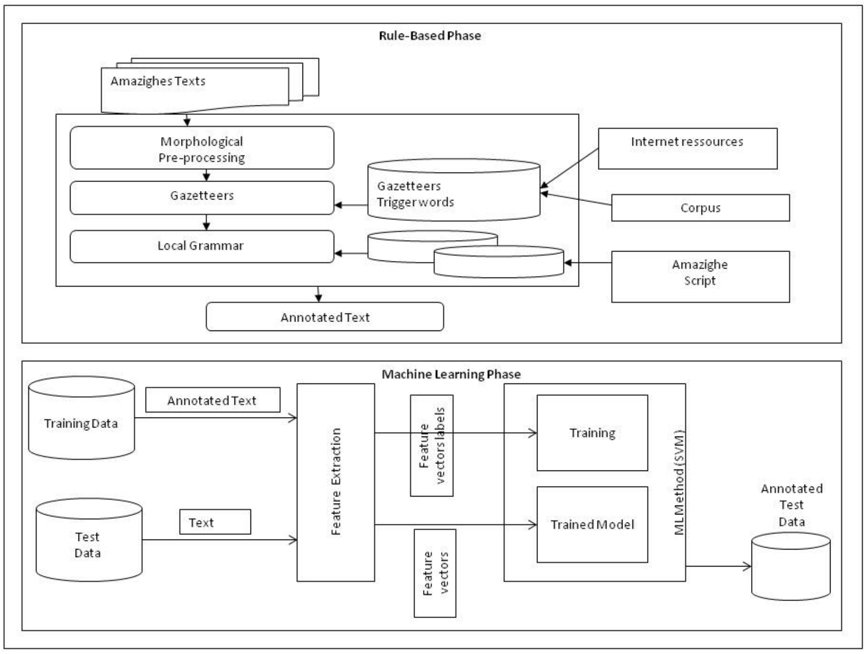
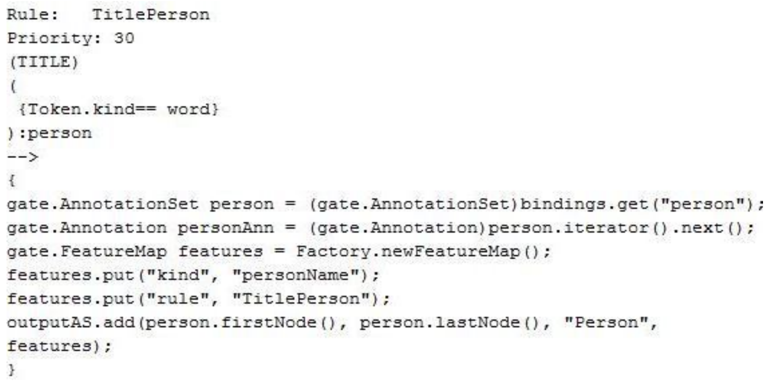


Fig. 2. Structure of our NER System

3.1 The Rule Based Phase

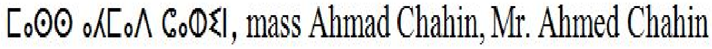
The rule-based component in our hybrid system is a reproduction of the NERAM system[15] using GATE[22] framework. The rule-based mode is developped with the abilility of recognizing the 5 NEs. The recognition process used contains two principal steps: a lookup procedure, called Gazetteers, including lists of known named entities; and a finite state transducer, called Grammar, based on a set of grammar rules derived by analyzing the local lexical context relieved from our corpus (examples is provided in Figure 3). We arrive at these resources after examining several sample news articles and try to make their coverage as high as possible.



```
Rule:   TitlePerson
Priority: 30
(TITLE)
(
 {Token.kind== word}
):person
-->
{
gate.AnnotationSet person = (gate.AnnotationSet)bindings.get("person");
gate.Annotation personAnn = (gate.Annotation)person.iterator().next();
gate.FeatureMap features = Factory.newFeatureMap();
features.put("kind", "personName");
features.put("rule", "TitlePerson");
outputAS.add(person.firstNode(), person.lastNode(), "Person",
features);
}
```

Fig. 3. Example rule for Person name recognition

This rule would be able to recognize a person name based on the trigger words. Example shown in Fig. 4 would be recognized by the previous rule.



ⵎⴰⵙⵙ ⴰⵃⵎⴰⴷ ⵛⴰⵃⵉⵏ, mass Ahmad Chahin, Mr. Ahmed Chahin

Fig. 4. Example Person Name Preceded by Person Title

The GATE environment is used to build the rule-based mode. Table 1 illustrates the number of gazetteers and rules implemented within each NE type. The system contains a total of 75 rules and 24 gazetteers.

3.2 Machine Learning Phase

The ML-based phase consists on two principal steps: feature extraction and selection of ML classifiers. The first step is the feature extraction which requires the selection of classification features. The features explored are divided into various categories:

Table 1. The Number of Gazetteers and Rules in each NE Type

Named Entity Type	Rules	Gazetteers entries
Person	16	2482
Location	15	2017
Organization	13	504
Date/Time	23	170
Numerical Expressions	8	152

Context words: These are the preceding and following words surrounding the current token, ie, these are the word set adjacent to NE. This feature accounts the different contexts in which NEs appear in the training data. All of these context relations and similar information can be collected as some useful features for predicting the unknown named entities. In our implementation, features are weighted according to their distance from the current instance annotation. In other words, features which are further removed from the current instance annotation are given reduced importance.

Gazetteers: This is the gazetteer feature, gathered from the look-up gazetteers: handcrafted lists of names of person names, locations (Countries, cities, ...), organization names (association, institutes, ...), date (hours, days, years, ...) and numerical expressions (numbers, percent, ...). This feature can be determined by finding a match in the gazetteer of the corresponding named entity type.

Mention: We prepared our corpus with annotations providing class information as well as the features to be used. Actually in GATE each class has its own annotation type (Date, Person, Organization, etc.), but the Machine Learning processing resource in GATE expects the class to be a feature value, not an annotation type. So, we have created a class information in the form of a single annotation type, named ”Mention”, which contains a feature ”class”.

The second step concerns the ML classifier used in the training, testing and prediction phases. The SVM ML technique has been chosen for their high performance in NER in general and Amazighe NER in particular.

In this work, GATE, an efficient workbench that support a large number of ML algorithms, is employed as the environment of the ML task.

4 Experimental Datasets

Amazighe Language suffer from the scarcity of language technological advancements. For NER in Amazighe language, suitable corpora have until recently been unavailable, thus we have created our own corpora, besides as we mentioned in previous works we have developed a stop word list, a triggers word list, and gazetteer component, that could be more helpful for our task. In this part we introduce our resources built for Amazighe.

4.1 Corpus and Sets used

Our aim was to set up a resource comparable to more traditional general corpus used for other languages, containing a wide range of text types and topics. We have built a large corpus of Amazighe language constructed by crawling the ”MapAmazighe”[21] website, which is the Amazighe information portal of ”Maghreb Arab Press(MAP)”, as well it is one of the largest freely available linguistic resources for Amazighe. The corpus contains more than 173 480 tokens. The corpus is actually a collection of 867 articles. Our goal was to construct a relatively heterogeneous topics, we have collected the whole news on royal activities of His Majesty King Mohammed VI (395 articles) and princely ones (93 articles), Regional (31 articles), Economics (58 articles), Social (60), Politics news (61), Sport (61), world activities (52 articles) and some general news (56 articles). We have decomposed our corpus into 4 sets, in order to minimize application execution times during the experiments. The sets ”1, 2, 3, 4” respectively contain around ”4168, 5273, 4963, 4281” distinct tokens. We manually annotate these data sets, using GATE that we used for this purpose, with MUC style named entity tags.

4.2 Evaluation data sets

We provide below statistical information regarding the evaluation data sets.

Set 1. The manual annotation lead us to a total of 6338 named entities. The annotated entities encompass 924 person, 1678 location, 332 organization names along with 582 date and 2822 numerical expressions.

Set 2. We preprocess this data set and the resulting set contains a total of 6827 named entities where 1452 of them are person names, 434 organization names, 1582 location names, 517 date and 2842 of them are numerical expressions.

Set 3. The manual annotation process results in the annotation of 6447 named entities with 1573 person, 1435 location, 287 organization names in addition to 744 date and 2408 numerical expressions.

Set 4. Similar to the previous data set, we obtained a total of 5039 named entities after annotation, with 936 person, 985 location, 416 organization names, 491 temporal expressions and 2211 numerical expressions.

5 Evaluation results and Analysis

In this section, we report the details of experimental setup, datasets of experiments and the evaluation results.

5.1 Metrics

In this work, we choose recall, precision and f-measure as three set-based measures . The definitions of recall, precision and F-Measure are given below:

$$Recall = \left(\frac{Correct + 0.5 * Partial}{Correct + Missing + 0.5 * Partial} \right) \tag{1}$$

$$Precision = \left(\frac{Correct + 0.5 * Partial}{Correct + Spurious + 0.5 * Partial} \right) \tag{2}$$

$$F - Measure = \left(\frac{2 * Recall * Precision}{Recall + Precision} \right) \tag{3}$$

In the preceding formulae:

- Correct corresponds to the number of named entities extracted by the system which are exactly the same as their counterparts in the answer key.
- Spurious represents the number of entities spuriously (erroneously) extracted by the system, they do not have corresponding annotations in the answer key.
- Missing is the number of named entities which are not annotated, hence missed, by the system although they are annotated in the answer key.
- Partial denotes the number of named entities extracted by the system which have corresponding entities annotated in the answer key with the same type, hence their type is correct but the tokens they contain are not exactly the same since either some tokens are erroneously missed or included by the system.

From the definitions, while recall tries to increase the number of tagged entries as much as possible, precision tries to increase the number of correctly tagged entries, and F-measure is the harmonic mean of recall and precision.

5.2 Results Obtained

The evaluation results of our system on these data sets are provided in table 2 using the above metrics. Results show that the rule-based approach leads to

Table 2. Performance of Our Rule-based System

Named Entity Type	Precision (%)	Recall (%)	F-Measure (%)
Person	98	100	99
Location	99	100	99
Organization	99	100	93
Date/Time	96	98	97
Numerical Expressions	71	87	79

good results. Apparently, Rule-based approach has best accuracy on categories of people, organization and localization as types of NE, but there are many discrepancies with the rest, this is due to the confusion that our system makes between Temporal and Numerical Expressions.

Table 3. Performance of Our System

Named Entity Recognition System	Precision (%)	Recall (%)	F-Measure (%)
Rule Based Approach	90	97	93
Hybrid (ML + Rule Based)	81	67	73

For the second experiment, we applied our hybrid system on our corpus, we splitted the corpus into training and test data, to truly know how well a machine learner is performing, for training we have selected 3 sets and 1 set for test phase.

Just to remain, we used the LibSVM SVM implementation. In this experiment, we used the linear kernel with the cost C as 0.7 and the cache memory as 100M. Additionally we used uneven margins, with τ as 0.4. The classification type is set as 'one-vs-others', meaning that the Machine Learning API will convert the multi-class classification problem into a series of binary classification problems using the one against others approach.

If we focus on results in table 3, we can easily deduce that our hybrid approach performed quite poorly in terms of precision, recall and f-measure, probably due to the nature of the dataset, distribution of our training and data sets, limited surrounding context, spelling mistakes, machine learning parameters and features used for this experiment and this clearly shows the necessity of determining appropriate feature set for the problem. Although it achieved good accuracy and we are currently working on expanding rules, testing more features to help in improving performance.

To summarize, all of the proposed systems achieve promising results on the test data set which is a meaningful contribution to NER research on Amazighe Texts, as related work is quite lacking compared to studies on other languages such as English, French, Chinese, etc., but to the best of our knowledge, our proposed system is the first to apply hybrid approach to NER on Amazighe texts.

Yet, we expect that the results should be verified on larger test corpora and can be improved by increasing the annotated training data set. Other crucial future task is to make a deeper elaboration of the employed parameters and features set to better evaluate their effects.

6 Conclusion & future works

Applying Named Entity Recognition for Amazighe language is a challenging, emerging research area, gaining more significance every day, especially due to the increase in the size of Amazighe texts that need to be processed, but nonetheless, building a NER system for Amazighe Language is still an open problem because it exhibits characteristics different from English. In this paper, Our hybrid NER system has the ability to enrich its lexical resources with those that it learns from annotated texts through learning approach. Both the hybrid system and its rule based predecessor are evaluated on 4 data sets of different genres: news on royal activities and princely ones, financial and social news texts, regional and politic news, sport and world activities texts and some general news. These data sets are manually annotated by the authors due to the lack of available annotated corpora for NER research in Amazighe language. The evaluation results shown that our proposed method achieves promising results, but the rule based approach still perform better than our hybrid approach.

Finally, this paper envisions possible improvements on the approach in order to further increase the score ot the proposed system, including larger annotated corpus, integrating POS tagging processing, deep analysis on features set (e.g. morphological features, etc) doing to experiment with varying the configuration file to see if we can produce varied results and appliying other machine learning mode to decide which one has the best performance on our data.

References

1. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes, vol. 30, no 1, p. 3–26. (2007)
2. Tan, J. K., Benbasat, I.: The Effectiveness of Graphical Presentation for Information Extraction: A Cumulative Experimental Approach*. Decision Sciences, vol. 24, no 1, p. 167–191. (1993)
3. Costantino, M., Morgan, R. G., Collingham, R. J., Carigliano, R.: Natural language processing and information extraction: Qualitative analysis of financial news articles. In Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997, pp. 116–122. IEEE (1997)
4. Feifan, L., Jun, Z., Bibo, L., Hao, Y., Yingju, X.: Study on Product Named Entity Recognition for Business Information Extraction. Journal of Chinese Information Processing, vol. 20, No. 1, pp. 7–13. (2006)
5. Leaman, R., Gonzalez, G.: BANNER: an executable survey of advances in biomedical named entity recognition. In Pacific Symposium on Biocomputing, vol. 13, pp. 652–663. (2008)
6. Mandl, T., Womser-Hacker, C.: The effect of named entities on effectiveness in cross-language information retrieval evaluation. In: Proceedings of the 2005 ACM Symposium on Applied Computing (SAC 2005), pp. 1059–1064. (2005)
7. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing, and retrieval. In The Semantic Web-ISWC 2003, pp. 484–499. Springer Berlin Heidelberg (2003)
8. Cimiano, P.: Ontology learning from text. pp. 19–34. Springer US (2006)
9. Jin, W., Ho, H. H., Srihari, R. K.: OpinionMiner: a novel machine learning system for web opinion mining and extraction. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1195–1204. ACM (2009)

10. Nobata, C., Sekine, S., Isahara, H., Grishman, R.: Summarization system integrated with named entity tagging and IE pattern discovery. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002). Spain (2002)
11. Pizzato, L.A., Molla, D., Paris, C.: Pseudo relevance feedback using named entities for question answering. In: Proceedings of the 2006 Australian Language Technology Workshop (ALTW-2006), pp. 89–90. (2006)
12. Babych, B., Hartley, A.: Improving machine translation quality with automatic named entity recognition. In: Proceedings of EAMT/EACL 2003 Workshop on MT and Other Language Technology Tools, pp. 1–8. (2003)
13. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. In: 16th international conference on computational linguistics, pp. 466–471. COLING (1996)
14. Talha, M., Boulaknadel, S., Aboutajdine, D.: NERAM : Named Entity Recognition for Amazighe language. In: 21th International conference of TALN. pp. 517–524. Aix Marseille University, Marseille (2014)
15. Boulaknadel, S., Talha, M., Aboutajdine, D.: Amazighe Named Entity Recognition Using a Rule Based Approach. In: 11th ACS/IEEE International Conference on Computer Systems and Applications. Doha, Qatar (2014)
16. Talha, M., Boulaknadel, S., Aboutajdine, D.: L'apport d'une approche symbolique pour le repérage des entités nommées en langue amazighe. In: EGC. pp. 29–34. Luxembourg (2015)
17. Chaker, S.: Textes en linguistique berbère - introduction au domaine berbère. éditions du CNRS. pp. 232–242. (1984)
18. Cohen, M.: Langues chamito-sÃĺmitiques. Edouard Champion, (1924)
19. Institut Royale de la Culture Amazighe, http://www.ircam.ma
20. Boukhris, F., Boumalk, A., Elmoujahid, E., Souifi, H.: La nouvelle grammaire de l'amazighe. IRCAM, Rabat (2008)
21. Amazighe Information Portal of "Maghreb Arab Press (MAP)", http://www.mapamazighe.ma
22. General Architecture for Text Engineering, https://gate.ac.uk/

Defining the Gold Standard Definitions for the Morphology of Sinhala Words

Welgama Viraj[1], Weerasinghe Ruvan[1], and Mahesan Niranjan[2]

[1]University of Colombo School of Computing,
No:35, Reid Avenue, Colombo 00700
Sri Lanka.
[2]University of Southampton
Highfield, Southampton,
SO17 1BJ, UK.
[1]{wvw,arw}@ucsc.cmb.ac.lk
[2]mn@ec.soton.ac.uk

Abstract. In this work, we describe the steps and strategies we carried out on defining morpheme segmentation boundaries of Sinhala words (which we called *Gold Standard Definitions*). We measured the coverage of the defined resource against three different Sinhala corpora and obtained over 70% coverage for each corpora. Then we report some interesting facts and findings about the Sinhala language revealed due to this development and finally about some applications of this valuable linguistic resource.

Keywords: Sinhala Morphology, Gold Standard Definitions, POS categories for Sinhala

1 Introduction

Identifying the morpheme boundaries of a word is very essential for modern Natural Language Processing tasks. It is the fundamental goal of any automatic morpheme induction algorithm or any rule-based morphological analyzer. The accuracy of identifying morpheme boundaries effects to the permanence of its applications such as Speech Recognition, Machine Translation, Information Retrieval and Statistical Language Modeling, specially if those are performed with morphological reach languages.

There are two major approaches for identifying morpheme boundaries of a word namely; *knowledge-based* approaches and *data-driven* approaches. Though very successful, the knowledge-based approaches are very expensive with respect to the human resource they require. As a result, research on morphological segmentation is now moving towards more data-driven approaches, which require less expertise and heuristics, but rely on data [1]. However, in order to precisely evaluate such data-driven approaches it requires a pre-defined morpheme definitions, referred to as *Gold Standard definitions.* Some key competitions on developing data-driven approaches such as Morpho Challenge Competition [2] have used gold standard definitions as one way of evaluating the algorithms and they have provided some sample Gold Standard definitions for English, German, Turkish and Finnish [3].

Our goal in this paper is to present the methodology and some findings on developing such resource for identifying morpheme segmentation boundaries of Sinhala words. Sinhala is an Indo-Aryan language spoken by more than 16 million people in Sri Lanka. Sinhala is a highly inflectional language as are many other Indic languages, and like many of them, can be considered as a low-resourced language with respect to the linguistic resources available for NLP. Therefore we assume that developing this kind of resource for Sinhala will provide a potential infrastructure for future research in Sinhala language. The rest of the paper describes the work carried out in detail.

2 POS Categories

Defining morpheme segmentation boundaries of words in a particular language is a highly challenging task, which needs lots of linguistic expertise and heuristic knowledge. Expert native speaker knowledge is required to classify words in to basic and sub POS categories . [4] have made some effort to define major POS categories of the Sinhala language and all the sub-structures of each category with a comprehensive list of words for each category. We used this work as the base for defining morpheme segmentation boundaries.

Having observing each POS category defined in [4], we decided to initially define morpheme segmentation boundaries only for five main POS categories namely; nouns, verbs, adjectives, adverbs and function words. [4] have introduced a novel sub classification for each of these categories according to their inflectional/declension paradigms and these subclasses are mainly specified by the morphophonemic characteristics of stems/roots.

2.1 Nouns

[4] have introduced 22 such sub categories for nouns based in their morphophonemic characteristics at the end of the word. We identified 26 sub categories based on their behavior in inflections and Table 1 shows all the sub categories defined for Sinhala nouns with number of words and number of inflected forms generate from each category with an example. [4] have identified 130 word forms for nouns in general, but we observed that non of these sub categories are inflected to all of these 130 forms.

As shown in the 4$^{\text{th}}$ column of the Table 1, masculine nouns generate the maximum number of inflected forms per sub category, which is 58. We classified 11,970 noun stems into these 26 sub categories and hence we were able to define morpheme segmentation boundaries for 529,781 distinct Sinhala nouns. The methodology we used to define these boundaries will describe later in this paper.

Table 1. Sub-categories for nouns

Group	**Subclass**	**Words**	**Forms**	**Example**
Masculine	FrontVowel. MidVowel	1,186	58	gɑwə(*cow*)
	Germinated Consonant	972	58	bɑlu (*dog*)
	BackVowel	190	58	elu (*goat*)
	Retroflex-1.1	48	58	kɑputu (*crow*)
	Retroflex-1.2	31	58	utumä (*lord*)
	Retroflex-2.1	19	58	kumərə(*prince*)
	Retroflex-2.2	37	30	sɑhɑkɑru (*partner*)
	Consonant-1	60	58	minis (*man*)
	Consonant-2	9	58	hɑrɑk (*bull*)
	Consonant-3	4	58	girä (*parrot*)
Feminine	FrontVowel. MidVowel	166	47	kuməri (*princess*)
	BackVowel	72	47	äryä (*lady*)
	Consonant	13	44	məw (*mother*)
Neuter	FrontVowel. MidVowel	4,234	42	mæ̈sə(*table*)
	Germinated Consonant	207	42	kɑju (*nuts*)
	BackVowel	1,070	42	putu (*chair*)
	Retroflex-1	122	45	siruru (*body*)
	Retroflex-2	519	45	irə(*sun*)
	Consonant	2,272	42	gɑs (*tree*)
	MidVowel	116	33	kɑdə(*shops*)
kinship	kinship-1	31	42	ɑkkä (*sister*)
	kinship-2	32	46	gurutumä (*teacher)*
	kinship-3	102	27	mɑllë (*brother*)
Uncountable	Consonant Ending	187	12	käbən (*carbon*)
	Vowel Ending	214	12	sëni (*sugar*)
Irregular	Animate	57	16	nönä (*lady*)

2.2 Verbs

Even though verbs are playing the most significant role of the meaning of a sentence, number of verbs in a particular language is far below than the number of nouns of that language. Hence, the classification of verbs into sub categories is simpler than nouns. [4] have identified 4 sub categories for Sinhala verbs, but we further divided one of this category into two by considering their behavior when generating inflected forms. Table 2 shows all the sub categories defined for Sinhala verbs with number of words and number of inflected forms generate from each category with an example.

As shown in the table 2, number of inflected forms of Sinhala verbs are much higher than nouns. The reason behind of this higher number of inflected forms for Sinhala verbs is the gerund forms (*verbal nouns*). There are 3 main gerund forms for each category and each of those forms are inflected to around 40 different forms as in nouns. All together there are 117 gerund forms for each sub category. However, some of these gerund forms are high frequency nouns. for example the word "godənægillə" (*the building*) is a high frequency noun and a general person may not be aware that it is derived from the verb "godənɑgənəwä

Table 2. Sub-categories for verbs

Subclass	Words	Forms	Example
ə-ending	487	206	bɑlə (*to see*)
e-ending	323	198	sinäse (*smiling*)
i-ending-1	47	200	rɑki (*to protect*)
i-ending-2	44	200	ɑndi (*to dress*)
irregular	108	-	bo (*to drink*)

(*to build*). We decided to consider these gerund forms as derivatives of verbs, but we can still consider them as nouns whenever necessary since we have tagged them as *gerund*. We identified 1,009 Sinhala verb roots in all 5 sub categories and coverage of it will be described later in this paper.

2.3 Adjectives

There are two main categories for adjectives. One is playing the adjectival role in a sentence based on its position while the other category is pure adjectives such as "usə" (*tall*) or "hondə" (*good*). Most of the time the noun stems play the adjectival role as in "putu kɑkulə" (*chair's leg*) or "minis hɑndə" (*human voice*). We only consider pure adjectives under this category and we identified 2,576 pure adjectives for Sinhala. All the adjectives are inflected for 2 forms and we named them as "*conjunction form*" (for example "hondɑtə" (*good and*)) and "*final form*" (for example "hondɑyi" (*is* good)).

2.4 Adverbs

As adjectives, adverbs can also be divided into two categories as *derivative adverbs* and *pure adverbs*. We only considered pure adverbs under this category and 245 such adverbs were identified. All the adverbs are also inflected for 2 forms as in adjectives.

2.5 Function Words

We identified 6 types function words for Sinhala. 4 of them were further divided into two groups as "*vowel endings*" and "*consonant endings*" and it helps to programmatically generate the corresponding inflected forms of each category. We identified 619 function words for Sinhala in all of 6 sub categories and Table 3 shows its distribution over each sub category.

Table 3. Sub-categories for function words

Group	Subclass	Words	Forms
Conjunctions	vowel endings	17	3
	consonant endings	12	2
Determinants	vowel endings	52	3
	consonant endings	46	1
Interjections	-	44	1
Particles	vowel endings	110	3
	consonant endings	35	2
Postpositions	vowel endings	107	3
	consonant endings	39	2
Verbparticles	-	157	1

3 Methodology

As described in section 2, we grouped all Sinhala words into 43 sub categories based on their POS categories and word endings. The main objective of this classification is to programmatically generate the morpheme boundaries for rest of all the words of each category based on a given definition file from each category.

3.1 Creating the Definition File

To define the morpheme boundary definitions for each category, we selected a word from each category and manually define all the morpheme boundaries for each of its inflected forms with help of native language experts. We defined two types of definitions for each word namely; "*definitions with morphs*" and "*definitions with features*".

Definitions with Morphs

In these definitions, we tried to define morpheme boundaries of a word based on its orthography and we did not consider the orthographic changes happening into the word ending when adding a suffix. We split a word into its morphemes and defined its form (the morph realization at the particular word) and the definition of the morpheme separately. The fundamental rule we kept on splitting morphemes is that they should be able to produce the relevant word by simply concatenating all the morphs. The objective of following such rule is to use these gold standard definitions to evaluate machine generated morpheme boundaries, which always split a word into morpheme like units based on its spellings. We used colon (:) to separate the morphs from its name and Table 4 shows a sample of these definitions for the Sinhala word "ගවයා" (*the cow*), which is from the category *Nouns-Masculine.FrontVowel.MidVowel.*

˜ stands for the empty morph which we used to denote hidden morphs of a word which is highly utilized in Sinhala Nouns.

Table 4. Example for definitions with morphs

Word	Definition
ගව (*cow-Root*)	ගව:ගව-N+RT
ගවයා (*the cow*)	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+NOM
	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+ACC
ගවයාත් (*and cow*)	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+NOM ත්:+CJ
	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+ACC ත්:+CJ
ගවයායි (*is cow*)	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+NOM යි:+FN
	ගව:ගව-N+RT යා:+SG ˜:+DF ˜:+ACC යි:+FN
ගවයෙක් (*a cow*)	ගව:ගව-N+RT ෙ◌ය:+SG ක්:+ID ˜:+NOM

Definitions with Features

In these definitions, we defined morphological features of a particular word with its root or stem. Table 5 shows a sample of these definitions with features for the same example as above.

Table 5. Example for definitions with features

Word	Definition
ගව (*cow-Root*)	ගව:ගව-N+RT
ගවයා (*the cow*)	ගව:ගව-N +SG +DF +NOM ගව:ගව-N +SG +DF +ACC
ගවයාත් (*and cow*)	ගව:ගව-N +SG +DF +NOM +CJ ගව:ගව-N +SG +DF +ACC +CJ
ගවයායි (*is cow*)	ගව:ගව-N +SG +DF +NOM +FN ගව:ගව-N +SG +DF +ACC +FN
ගවයෙක් (*a cow*)	ගව:ගව-N+RT +SG +ID :+NOM

The objective behind this definition is to use this gold standard definitions as a resource for a *Sinhala Morphological Analyzer / Generator*. However, this is a derivative work of the *definitions with morphs*, but we kept defining this separately for the simplicity.

3.2 Creating the Roots File

The lexicon defined by [4] were used to create list of roots for a particular category. For some categories, the root form changes when adding some inflectional suffixes. For example, the verb root “bɑlə” (*to see*) become “bælæ” when adding inflectional suffixes for past tense. These changes can not be automatically predicted for some categories and hence we manually compiled the roots files with these alternative forms for each category.

After defining morpheme boundaries for all inflectional forms of a selected word in a category as described above, a computer program used to generate those definitions for all the other words of its category. The program requires the definition file and the list of roots of a particular category and it replaces the definition file’s root with other roots. This approach helped us to generate such morpheme boundary definitions for most Sinhala words with less effort.

4 Statistics

We managed to compile the first version of Sinhala Gold Standard Definitions with 736,084 Sinhala words using the above approach. Table 6 shows the number of root forms covered in each POS category with their percentage with respect to the total number of stems.

Table 6. Distribution of number of stems of each POS category

Category	**No. of Stems**	**%**
Nouns	11,971	72.90
Verbs	1,009	6.14
Adjectives	2,576	15.69
Adverbs	245	1.49
Function Words	619	3.77
Total	**16,420**	**100.00**

As shown in the Table 6, nouns cover nearly 73% of stems. That is expected because nouns are the most common POS category of a language. However, it is interesting to see that the number of adjectives in Sinhala is much higher than number of verb roots. This phenomenon is changed when we consider the total number of words of each category including their inflected forms. Table 7 shows number of total word forms of each category with their percentage with respect to the total number of words.

It is interesting to see that the coverage of nouns with respect to the total number of defined words are almost similar as it in stems. However, the percentage of verb forms of the language is significant with respect to the other categories other than nouns. As shown in the Table 7, number of adverbs in Sinhala is negligible with compare the total number of words of the language.

Table 7. Distribution of number of total words of each POS category

Category	**No. of Words**	**%**
Nouns	529,781	71.97
Verbs	196,873	26.75
Adjectives	7,503	1.02
Adverbs	671	0.09
Function Words	1,256	0.17
Total	**736,084**	**100.00**

5 Coverage

We measure the coverage of this defined resource against 3 different Sinhala corpora. The main resource we used to measure the coverage is the UCSC 10M words Sinhala corpus described in [5] and 70% of the 10 million words are covered by the defined resource. Interestingly, the coverage against the unique word list extracted from the above corpus is only 20.64% and that gives a clue on applicability of Zipf's law for the Sinhala language. We figured out that the most of uncovered words are proper nouns (which are not covered by the defined resource) and typos.

Second resource we used to measure the coverage of the defined resource is 2.4M words Sinhala news corpus extracted from online newspapers. The coverage of the defined resource is 72.65% against this news corpus and it is slightly better than 10M words open domain corpus. The reason behind this slight improvement may be due to less number of typos in online newspapers.

We used 0.95M words Sinhala news editorial corpus as the third resource for measuring the coverage and we obtained 78.27% coverage against the editorial corpus. It can be assumed that the low number of proper nouns and professional writing styles of newspaper editors are the reasons for this improvements.

6 Applications

One of the main objective of developing such resource is to use these definitions to evaluate machine learning approaches on automatic morpheme boundary detections. This resource can directly be used to check the accuracy of the output of such approaches and it will give a precise measure on the performance of morpheme induction algorithms other than any other evaluation methods. Such attempt has described in Kurimo et al. (2010).

Another direct application of such resource is a rule-based morphological analyzer for the particular language. We developed a rule-based morphological analyzer for Sinhala using this resource and that is the first such tool available for Sinhala. Currently, this morphological analyzer is using for research on Sinhala speech recognition and Sinhala-Tamil machine translation and the results of them are yet to be published. Other than serving as a language resource for NLP research in Sinhala, this resource is expected to be used as a learning material for Sinhala.

7 Conclusion

We presented the approach and the data sources we used to develop the Gold Standard Definitions on marking morpheme boundaries for Sinhala words. The defined resource covered over 70% open domain Sinhala words. This is the first attempt on define such resource for Sinhala language and we hope that this resource will be useful for many NLP applications for Sinhala in the future.

Acknowledgments

The authors are very grateful to past and current members at the Language Technology Research Laboratory of the University of Colombo of School Computing, Sri Lanka for their significant contribution on developing basic linguistic resources for Sinhala language such as corpora and lexica, which make it possible for us to carry out the work described above.

References

1. Welgama V., Weerasinghe R., and Mahesan M.: Evaluating a Machine Learning Approach to Sinhala Morphological Analysis. In: Proceedings of the 10th International Conference on Natural Language Processing, Noida, India (2013)
2. Morpho Challenge 2010 - Semi-supervised and Unsupervised Analysis, `http://research.ics.aalto.fi/events/morphochallenge2010/`
3. Kurimo M., Virpioja S., Turunen V., and Lagus K.:Morpho Challenge 2005-2010: Evaluations and Results. In: Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, pp. 87–95. Association for Computational Linguistics, Uppsala, Sweden. (2010)
4. Weerasinghe R., Herath D., and Welgama V.: Corpus-based Sinhala Lexicon. In: Proceedings of the 7th Workshop on Asian Language Resources, pp. 17–23. Association for Computational Linguistics, Singapore (2009)
5. Weerasinghe R., Herath D., Welgama V., Medagoda N., Wasala A., and Jayalatharachchi E.: UCSC Sinhala Corpus - PAN Localization Project-Phase I. Language Technology Research Laboratory, University of Colombo School of Computing (2007)

Attachment Errors of Nouns after Possessor Clitic

Ercan Solak[1], Olcay Taner Yıldız[1], Onur Görgün[1,2], and Razieh Ehsani[1]

[1] Işık University, Istanbul, Turkey
[2] Alcatel Lucent Teletaş Telekomünikasyon A.Ş., Istanbul, Turkey

Abstract. In this paper, we analyze the errors of NP attachments that occur when they combine with NP's ending in possessor clitic. We suggest a simple pattern based detection and correction solution. We illustrate the errors with examples from Penn Treebank.

1 Introduction

With the help of the annotated corpora of different size and nature, statistical NLP research has come a long way over the past three decades. Part-of-speech tagged Brown corpus [1], syntactically annotated Penn-Treebank [2] and sentence aligned Europarl corpus [3] are arguably the most-cited corpora in their respective subfields of statistical NLP research. Creating and maintaining such huge corpora need a lot of manual and semi-manual effort. The annotation errors may have a negative influence on the performance of the corpus-based research.

To overcome this problem, several error detection and/or correction algorithms are proposed [4]. Early works concentrate on the detection of errors in the POS-annotation [5], and they were followed by the works focused on the detection of errors in the syntactic structure [6]. Not only the users of the corpora, but also the maintainers of the corpora are also interested in detecting annotation errors. For example, Linguistic Data Consortium uses Tree Adjoining Grammar (TAG) to check treebank consistency, which is applied on Arabic and English treebank [7].

In this paper, we focus on the attachment errors that occur when NP attachments are combined with the possessor clitic. We give examples and statistics from the Penn Treebank. We propose a simple correction algorithm based on the specific pattern of the error.

This paper is organized as follows: We give the related work on annotation error detection/correction in Section 2. After reviewing very briefly the attachment errors in Section 3, we will discuss the relationship between NP and PP, and show how this relationship can cause errors in the annotation. We illustrate the errors with example trees from Penn Treebank in Section 5 and illustrate its implications for tree based translation. We describe an error correction algorithm in Section 6. Finally, we conclude in Section 7.

2 Related Work

[8] is one of the earliest works in the annotation error detection. They divide corpus errors into three; (i) detectable errors which can be automatically detected and fixed,

pp. 173–181; rec. 2015-01-25; acc. 2015-02-27

(ii) fixable errors which require human intervention at some point in the correction process, and (iii) other cases, where the markup guidelines do not give any hint to the annotators and leave them to their own intuitions.

In general, annotation error literature can be divided into two; error detection approaches and error correction approaches. [9, 10, 5] detect errors in the POS annotation; [11–14, 7] detect errors in the constituent structure; [6, 15] detect errors in the dependency structure in the treebanks. On the other hand, [9, 16] can correct POS annotation errors; [4] can correct errors in the constituent structure.

3 Attachment errors

Attachment ambiguity is a source of error for automatic constituent parsers that rely on structural information. A common example is the sentence “I saw the man with a telescope”. The PP “with the telescope” can attach either under the NP “the man with a telescope” or under VP “saw the man with the telescope”. Obviously, a bit of world knowledge often resolves the ambiguity. When we replace “telescope” with “suitcase”, it is clear where the attachment goes.

4 NP and PP

Consider the sentence “I read John’s book of quotations.” The Berkeley parser [17] gives the parse tree given in Figure 1.

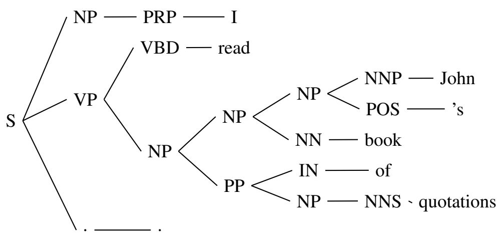


Fig. 1. The parse of the sentence “I read John’s book of quotations.”

The PP “of quotations” clearly qualifies the book that happens to belong to John. Now consider the alternative parse in Figure 2.

Again, “of quotations” qualifies the the book. But this time, parse tree attaches “book” with the PP first, before attaching the resulting NP subtree to “John’s”. The difference between the two cases seems minor. Indeed, in both cases, John has a book and the book has lots of quotations in it.

Let us slightly change the sentence while trying to keep the same semantics. “I read the book of quotations of John.” So, we got rid of the possessive clitic and tried to express the sentence with a uniform use of “of”s. But this introduces a genuine ambiguity.

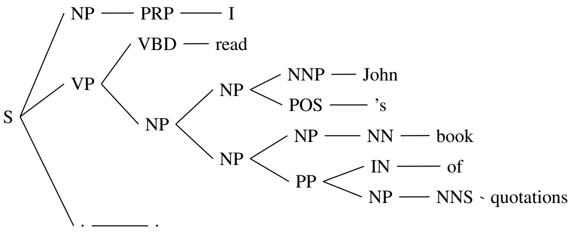


Fig. 2. An alternative parse of the sentence "I read John's book of quotations."

Does John have a book that has quotations by many famous people? Or is this a book that contains John's sayings exclusively?

Interestingly, both Berkeley and Stanford parsers [18] think the latter interpretation is more likely, although they slightly disagree on whether to use VP→VBP NP or VP→VBP NP PP. The Berkeley parse for this reading of the sentence is given in Figure 3.

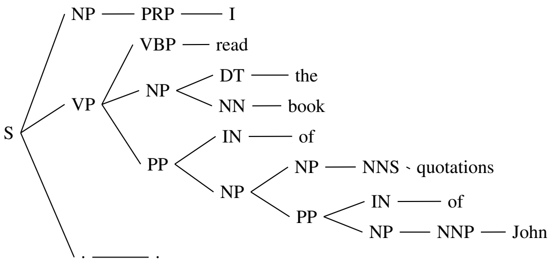


Fig. 3. The most likely Berkeley parse of the sentence "I read the book of quotations of John."

Thus, there are two opposing tendencies in the parsers for NPs expressing possession. When the possession is expressed with a clitic "'s", the parser tries to attach the next noun under the running NP. When "of" is used, the parser tries to group what comes after "of" into a NP first.

5 Penn Treebank

In this section, we focus on the case that has clitic. The incorrect constituent structure has a three layer template given in Figure 4. A,B and C are nouns and d is a preposition.

In order to see the frequency of the incorrect attachment in manual annotation, we searched for the incorrect template in Figure 4 among the 43908 sentences in the Penn Treebank. 614 of those fit the incorrect template. That is 1.4%, fairly high as the attachment errors go.

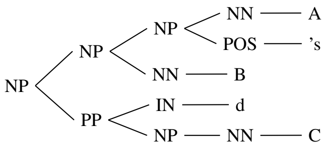


Fig. 4. Template tree structure of incorrect noun attachment after the possessive clitic.

6 Implications for translation

We noticed this NP attachment ambiguity when we were building a parallel treebank between Turkish and English [19]. In constructing the parallel corpus, we used a subset of trees in the Penn Treebank and translated them into Turkish using a sequence of only two operations. One operation permutes the children of a node and the other replaces a leaf with a stem or a morpheme.

Given the regularity of constituent order in English sentences and the regularity morphotactics of Turkish, it is possible to translate many sentences between the two languages using only the permute and replace operations.

An example translation is given in Figure 5.

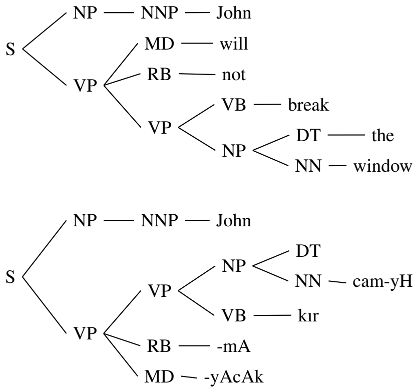


Fig. 5. The permuting of the nodes and the replacement of the leaves by the stems or morphemes.

Turkish uses postpositional morphemes to construct genitive-possessive noun phrases. In the construction A-GEN B-POSS, where A and B are nouns, A is the possessor and B is the entity possessed by A. GEN and POSS suffixes agree in their person markers. GEN-POSS construction can be chained as A-GEN B-POSS-GEN C-GEN. In this chain, A possesses B and B possesses C.

When we translate English trees to Turkish using only permutation and replacement, both the clitic "'s" and the preposition "of" is replaced with -GEN morpheme. -POSS morpheme is added to the possessed NP. Thus, food example, we have

(1) kapı-nın kol-u
door-GEN handle-POSS
handle of door

The trees for this pair are given in Figure 6.

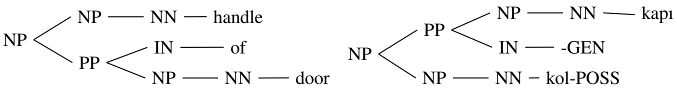
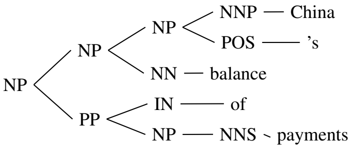


Fig. 6. Parse trees for the sentence pair in (1).

For an error prone example, consider the following NP subtree taken from the parse of the sentence "With less capital coming in, China's balance of payments will suffer".



Semantically, "of payments" qualifies the "balance" as there is no entity "China's balance" that can stand on its own. However, in the parse tree, the annotators preferred an early attachment and created this entity. For correct literal translation of this NP to Turkish we have,

(2) Çin-in ödemeler-i-nin denge-si
China-GEN payments-POSS-GEN
balance-POSS

When we replace the functional words the English tree with their Turkish morphemes only, we obtain the following tree.

Clearly, there is no way this tree can be permuted to read

(3) China-GEN payments-POSS-GEN
balance-POSS

If the original tree had the correct attachment, however, it could be easily permuted to the tree in Figure 7 with the correct order.

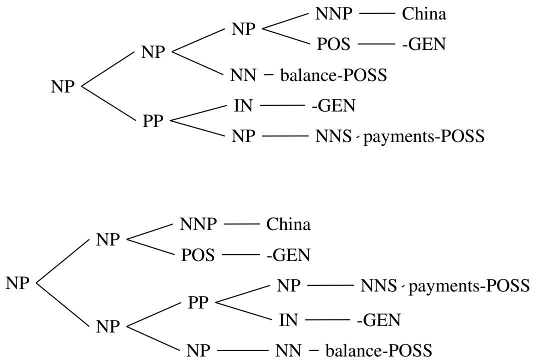
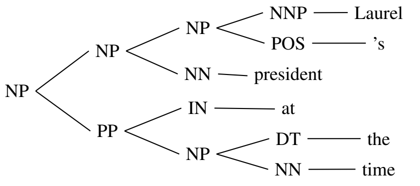


Fig. 7. Correct subtree of sentence (3).

We give two more examples of NP phrases from the Penn Treebank with their correct literal Turkish translations. These translations can not be obtained from the English tree through permutation and replacement. Note that PPs in these examples contain prepositions "at" and "for".

(4) Laurel-in o zaman-ki başkan-ı
Laurel-GEN that time-REL
president-POSS
Laurel's president at the time.



(5) Girişim-in Thomson için önem-POSS
Venture-GEN Thomson for
importance-POSS
The venture's importance for Thomson

7 Remedy

We can detect with a template the particular class of attachment errors we analyzed in this work. In order to correct the relevant subtree, we need to identify and move parts of it. The general incorrect pattern is given in Figure 8.

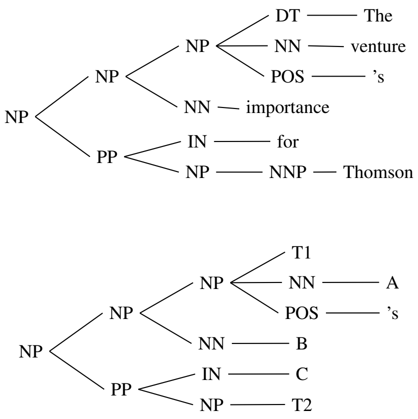


Fig. 8. Pattern for incorrect noun attachement.

Here, T1 and T2 denote nodes possibly with their own subtrees. A, B and C are terminal symbols. In order to correct the attachment error, the constituents T1, T2, A, B and C must be moved around such that now the tree has the structure in Figure 9.

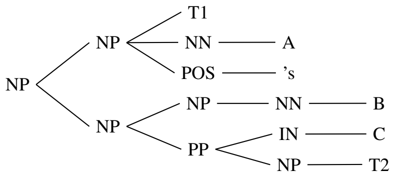


Fig. 9. Corrected tree.

Comparing the Figures 8 and 9, we see that the correction changes the counts of rules that contain NP on the left hand side. It increases the counts of NP→NP NP and NP→NN and decrease the count of NP→NP NN. We observed that this creates a tendency to introduce spurious NP hierarchies.

8 Experiment

To see the effect of these attachment errors on the parser performance, we trained the Stanford parser with the original and corrected data from WSJ section of the Penn Treebank. When we use original data for both training and test, we get an F score of 85.53. When we correct the attachment errors in both the training and test data, the F score becomes 85.4.

Interestingly, even though we corrected the attachment errors in the training treebank, the parser still consistently makes the attachment error in the test set. This is probably due to the tendency of the parser to attach NN subtree as a child to the preceding NP rather constructing a new NP above NN. The changing counts after the correction do not seem to be enough to reverse this tendency.

9 Conclusion

Phrase attachment ambiguities are sources of parse errors as well as funny newspaper clippings. In many cases, the ambiguity can be resolved using contextual and lexical constraints. In other cases, the errors are regular and follow a pattern.

In this work, we analyzed a particular class of noun phrase attachment errors. We found that this error occurs in about 1.4% of the sentences in the Penn Treebank. We showed an implication of the error for tree based translation. Finally, we suggested a template to detect the error in the Treebank and a simple rearrangement to put the constituents in their proper places. The score with the correct data is slightly lower.

References

1. Francis, W.N., Kucera, H.: Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US (1979)
2. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics **19** (1993) 313–330
3. Koehn, P.: Europarl: A multilingual corpus for evaluation of machine translation. (2002)
4. Kato, Y., Matsubara, S.: Correcting errors in a treebank based on synchronous tree substitution grammar. In: Proceedings of the ACL 2010 Conference Short Papers. ACLShort '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 74–79
5. Dickinson, M., Meurers, D.: Detecting errors in part-of-speech annotation. In: 10th Conference of the European Chapter of the Association for Computational Linguistics, The Association for Computer Linguistics (2003) 107–114
6. Boyd, A., Dickinson, M., Meurers, D.: On detecting errors in dependency treebanks. Research on Language and Computation **6** (2008) 113–137
7. Kulick, S., Bies, A., Mott, J.: Further developments in treebank error detection using derivation trees. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, European Language Resources Association (ELRA) (2012)
8. Blaheta, D.: Handling noisy training and testing data. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 111–116
9. Eskin, E.: Detecting errors within a corpus using anomaly detection. In: 1st North American chapter of the Association for Computational Linguistics Conference. (2000) 148–153
10. Nakagawa, T., Matsumoto, Y.: Detecting errors in corpora using support vector machines. In: 19th International Conference on Computational Linguistics. (2002) 709–715
11. Dickinson, M., Meurers, W.D.: Detecting inconsistencies in treebanks. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), Växjö, Sweden (2003) 45–56

12. Ule, T., Simov, K.: Unexpected productions may well be errors. In: 4th International Conference on Language Resources and Evaluation, European Language Resources Association (2004) 1795–1798
13. Dickinson, M., Meurers, W.D.: Prune diseased branches to get healthy trees! How to find erroneous local trees in a treebank and why it matters. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), Barcelona, Spain (2005)
14. Kulick, S., Bies, A., Mott, J.: Using derivation trees for treebank error detection. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 693–698
15. Volokh, A., Neumann, G.: Automatic detection and correction of errors in dependency treebanks. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 346–350
16. Murata, M., Utiyama, M., Uchimoto, K., Isahara, H., Ma, Q.: Correction of errors in a verb modality corpus for machine translation with a machine-learning method. ACM Transactions on Asian Language Information Processing **4** (2005) 18–37
17. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. (2006) 433–440
18. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. (2003) 423–430
19. Yıldız, O.T., Solak, E., Görgün, O., Ehsani, R.: Constructing a Turkish-English parallel treebank. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, Association for Computational Linguistics (2014) 112–117

An Unsupervised Text Normalization Architecture for Turkish Language

Savaş Yıldırım and Tuğba Yıldız

Department of Computer Engineering, Faculty of Engineering, Istanbul Bilgi University
Santral İstanbul,Eyüp, 34060 Istanbul, Turkey
savasy@bilgi.edu.tr
tugba.dalyan@bilgi.edu.tr

Abstract. A variety of applications on the problem of short-text messages require text normalization process that transforms ill-formed words into standard ones. Recently, many successful approaches have been applied to text normalization especially for social media text. Since each natural language has its own difficulties and barriers, we need to design an architecture to normalize short text messages in Turkish language which has an morphologically rich agglutinative structure. The model proceeds from simple solutions towards more complicated and sophisticated ones to reduce time complexity. A variety of techniques from lexical similarity to n-gram language modeling have been evaluated by exploiting several resources such as high quality corpus, morphological parser and dictionaries. We demonstrate that unsupervised text normalization architecture adapting both lexical and semantic similarity for Turkish domain has shown efficient results that might contribute to other studies.

Key words: lexical normalization, short text message, microblog, text preprocessing

1 Introduction

Micro-blogging social environment provides large volume of data on which many applications with regard to natural language processing (NLP) can easily be applied. According to reliable resources [1], %75 percent of Internet users are also active social media users. *Facebook* has over 1 billion users, *Twitter* has about 500 million ones. Moreover this trends are dramatically increasing over time. It is claimed that 30 billion devices will connect to Internet in the next decade. That creates an huge amount of social data, namely big data, and gives both challenge and possibility to the scientist and experts in all fields.

Usage of Internet eventually shapes the communication style and more importantly use of language. Texting Language is a new phenomenon for the field of NLP. Significantly, new generation starts to build their own language codes,

[1] http://www.techradar.com/news

as in Internet slang. Recently many NLP studies have been on Micro-blogging social media and texting language. Disaster detection, sentiment analysis, event discovery are among the most popular studies. In texting language, the quality of the language can vary from slang language to high-quality text. Therefore text processing tool requires a reliable normalization phase to understand the meaning of the messages. Short-text messages highly include typos, abbreviation, acronym, phonetic substitution, number substitution, use of interjection etc. All these raise an important barrier to be solved. Text normalization process is, thus, more important step in the preprocessing phase of the many studies.

There are a variety of studies regarding normalization especially in the most-studied languages such as English, German, and Spanish. Our primary motivation is to apply text normalization approach to a morphologically rich language, Turkish. Although it is one of the less-studied languages and its language resources are quite limited, Internet usage in Turkey is significantly higher than expected, which necessitates such preprocessing tools then. Each language has its own challenges due to grammar, lexicon, culture, and also political environment. For instance, some phonetic substitution such as number substitution as one of the best phenomenon in English, is not widely used in Turkish language. Having a rich morphological structure of a language turns out the problem a bit harder for our case.

With various experiments, we observed the structure of Texting Language in Turkish and we figured out many phenomena. Our normalization phases are based on those findings. Sufficient volume of both Twitter data and newswire corpus are mined to extract the rules, to conduct the experiment and to test. After morphological analysis of a given tweet, the system works on ill-formed tokens. A corpus and a pre-defined dictionary are exploited to build a look-up dictionary, namely IV, since that using a simple traditional dictionary cannot match many cases especially for agglutinative languages and special domains. The frequent terms occurring in the corpus are added to the IV list. A variety of assumptions are experimentally compiled and applied. Furthermore lexical similarity functions are used to correctly normalize the ill-formed tokens. Some cases such as missing a letter, using ASCII counterpart of a letter are easily recovered. Finally, an n-gram language model utilizing corpus evidence are applied to improve system performance. The architecture is designed in a manner of increasing degree of time complexity

2 Related Works

A variety of sources such as tweets, SMS messages, blogs, etc. have been used for normalization with different approaches; machine translation models, dictionary-based approach, language models, finite state transducers, and cascaded methods. One of the important attempts [1] proposes syntactic normalization with combining two steps; they first process the tweets to remove noise and feed them with statistical machine translation (SMT). Other studies have also used MT models for normalization [2–4].

Another approach which is an unsupervised model, with using noisy channel model, is presented by [5, 6] for normalization. The study [7] designs a normalization system that integrates enhanced letter transformation, visual priming, and string/phonetic similarity for SMS and Twitter data sets.

The study [8] proposed rule-based approach for normalization task. In [9], parser-centric view of normalization is presented. In study [10], a classifier is proposed to detect ill-formed words and produce their canonical lexical forms in standard English based on morphophonemic similarity. Word similarity and context are used to select the correct candidate. In [11], they present that dictionary-based approach outperforms state-of-the-art performance. The extension of these studies is proposed with significantly expanded experimentation in [12]

Although Tweet normalization have been widely and generally applied for English, other languages such as German, Spanish, Malay, French etc. have been also supported [13–18]. In Turkish, a cascaded approach has been applied to short text messages [19]. They propose the model by categorizing the problem into seven steps: letter case transformation, replacement rules & lexicon lookup, proper noun detection, deasciification, vowel restoration, accent normalization and spelling correction.

3 Methodology and Experimental Setup

3.1 Social Media Language

As social media continues to ruin the language, the normalization task is getting challenging. It might share the same problems with spell checking but they differ in that lexical variants in short text messages are often intentionally generated due to the need to save characters, for social identity, or due to the some convention. To normalize an ill-formed word, first we build an in-vocabulary (IV) list. If a word is both not a member of IV and not parsed by a parser, it is considered as ill-formed words.

For building a look-up IV, we exploit some resources; a morphological analyzer, a dictionary and a big reliable corpus. Under some circumference, morphological analyzer cannot distinguish correct words from ill-formed words. Some proper nouns cannot be handled by the parser then. Some trendy places, organizations, actors cannot be either detected by morphological parsed or found in standard dictionary. To build a broader dictionary, we exploit a reliable corpus that are mostly compiled from news texts that have been supervised or edited by experts. It consists of acceptable level of formal language. Most frequent terms or correctly parsed terms in it are retrieved and considered a reliable IV terms. Therefore almost all surface forms of a word can be represented by IV. A high coverage of IV might reduce the OOV rate as in all other morphologically rich language such as Fin, Turkish.

In study [20], they discuss the need for vocabulary size. For a comparison, they checked a English corpus consists of texts from the New York Times magazine. Whereas there are fewer than 200.000 different word forms in the 40

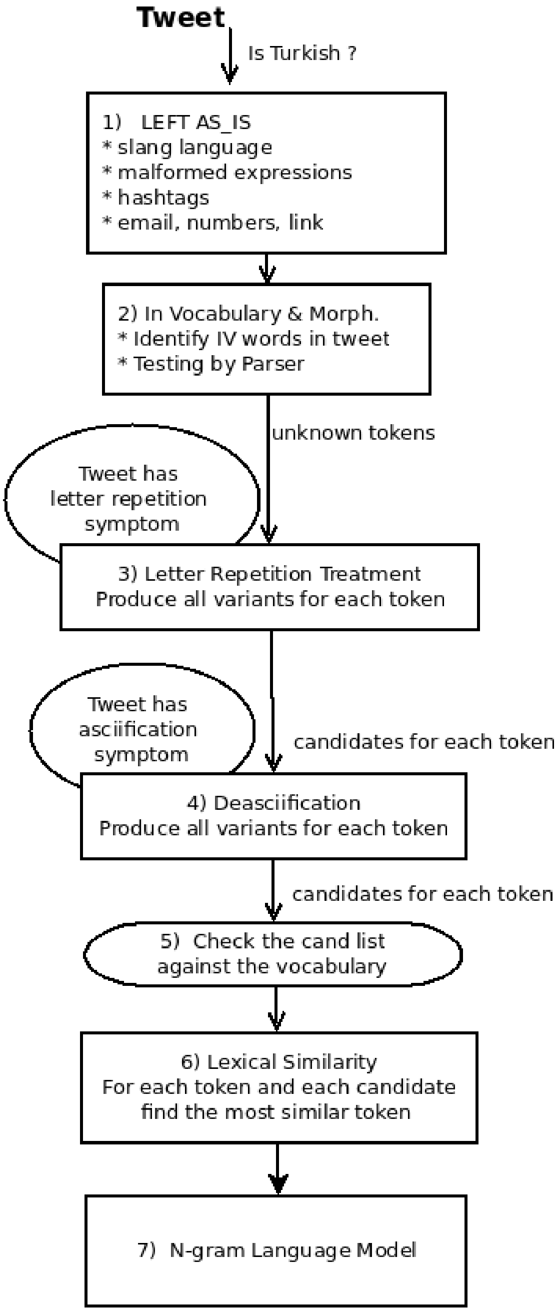

Tweet
Is Turkish ?
1) LEFT AS_IS
* slang language
* malformed expressions
* hashtags
* email, numbers, link
2) In Vocabulary & Morph.
* Identify IV words in tweet
* Testing by Parser
unknown tokens
Tweet has letter repetition symptom
3) Letter Repetition Treatment
Produce all variants for each token
Tweet has asciification symptom
candidates for each token
4) Deasciification
Produce all variants for each token
candidates for each token
5) Check the cand list against the vocabulary
6) Lexical Similarity
For each token and each candidate find the most similar token
7) N-gram Language Model

Fig. 1. Architecture of Normalization system

million word English corpus, the corresponding values for *Finnish, Estonian, Turkish* corpora of the same size exceed 1.8 million, 1.5 million and 1 million words, respectively. When we build a broader vocabulary, we reach the 2.4 M unique tokens out of 500 million token sized corpus from which all surface forms of the words are taken.

If a given token is not in IV and cannot parsed by parser, many steps are evaluated to detect the problems and to find a solution. However, we left some particular word untouched as listed below.

- By a slang dictionary with the size of ~200, all slang words are left as-is.
- 1 or 2 sized words are also left.
- a word mixed with numbers such as *tr12* are left.
- all hash-tags
- all e-mails, all numbers, digits are detected and left
- tokens containing some non-latin5 characters
- all emoticons are left as-is

All remaining words are passed to the next phases of system. According to unidentified words, we figured out some phenomena in Turkish Texting Language. This depicts how social media users use their language to communicate. The facts are as follows:

- Number substitution such as *2nite/(tonight)* as in English is not a case for Turkish Texting Language. Just few cases can be observed for the phonetic substitution.
- Intentionally using ASCII character set is widely spread due to save time and the reason preventing the message uncorrupted in other system that does not support *latin-5* character set. Turkish has own accented character set **(çöşüığ)**. In the mobile phones, to type **s** instead of **ş** is more easier. ASCII character set is widely used than Turkish accents. Replacing ASCII characters by their Turkish counterpart are called *deasciifying.*
- Clipping or shortening the messages to save time. *(Geliyorum− >geliyom, gelyom : koşacağım− > koşcam, koşcm)*
- The Turkish language encompasses a diverse range of accents and dialects as in other languages. This provides a wide range of pronunciation. Those pronunciations and dialects can reshape short text messages and their style. Writing in short text is akin to spoken language. People want to text as if they talk.
- *Interjection* are frequently used as in many languages. The words are lengthened by repeating the letters. When some syllabus is needed to stress, the repetition is used then.
- The suffix (-da) has two kind of usage in the grammar. First is to code locative case. Second meaning is *"also": "Ben de istiyorum" (I also want to).* In the latter case, a space must intervened between the word and the *da/de* suffix as in the example. Same holds true for the suffix *ki (the meaning is "that is").* In both cases, users can miss or do not care putting space between words. Therefore the word is left in the unidentified status and cannot be captured by the both morphology analyzer and the vocabulary.

3.2 Normalization Process

We develop different solutions for some of the problems listed above. The Figure 1 depicts high-level architecture of the system. In the *left-as-is* process, some tokens are left as it is written and tagged by the system. Slang terms, numbers, emails and other type of tokens are simply detected. Remaining ill-formed tokens are passed the next process. The further treatment processes are letter repetition, asciification, lexical similarity and n-gram modeling. After each treatment, token is checked against the IV. All the phases are ordered by their degree of complexity.

3.3 Letter repetition treatment

We generate a set of variants by varying the repetitive letters for a given token. For instance, for a token **“gitttikkkkk” (we have gone)**, after reducing the repetition number of a repetitive letter two as in *(gittikk)*, we produce all variants as in the list **(gittik, gitikk, gittik, gitik)**. 2^n variants are produced for a given token that includes n different repetitive letters. In this case the term **“gittik”** is found in IV and accepted. If more than one candidates captured in variant list, decision is made by checking term frequency. This solution is applied in each phase. If none of them is not in IV, we apply lexical similarity to find the closest IV word to any of those variant list.

3.4 Turkish Accents vs. ASCII counterpart

Most short messages have this phenomena since that it is more easy to type ASCII counterpart of the Turkish accents. Some studies called that phenomenon as *asciification.* So the solution is called *deasciification.* The studies immediately apply this approach for any given potential token. However, before applying this process, we need to detect a tweet has this kind of asciification sympthom or not. Instead of working on individual words, we would look entire tweet instead. Our assumption is that if a user types only ASCII letters, s/he does not use any Turkish accents in any position, then deasciification process is needed and applied. Likewise users who types Turkish accent tends not to asciify at all. If a token does not contain any Turkish accent and potential ASCII counterpart of Turkish accents, it is considered suspicious tweet in terms of asciification. This approach dramatically reduces program complexity and prevent applying unnecessary operations and wasting time. The computed probability of that a proper tweet contains any Turkish accent is 92%, where a proper tweet refers to contains at least n tokens, where n is 4. It is, then, said a Tweet (n>=4) is in Turkish language, it most likely contains Turkish accents.

Turkish characters and their ASCII counterparts are *“ğüşıöç”* and *“gusioc”*. All variants are produced for a given word. The ill-formed token *“cocuk” (kid)* contains four potential ASCII character set *(c,o,c,u)* that are intentionally replaced for Turkish accents. And variants of the words are *['cocuk', 'cocük', 'coçuk', 'coçük', 'cöcuk', 'cöcük', 'cöçuk', 'cöçük', 'çocuk', 'çocük', 'çoçuk', 'çoçük',* *'çöcuk', 'çöcük', 'çöçuk', 'çöçük']*. It has four suspicious ASCII characters and, 2^4, 16 variants are produced. In this list only the term *"çocuk"* is in IV. If more candidates are in the list, we select them according to their term frequency scores. If any IV word is found, lexical similarity is used between each variant and IV list.

Another crucial problem is that tweets have both asciificaiton and letter repetition problem together. Problems are cascaded and multiplied then. We apply nested procedure to reach normalized form. Again, remaining phases are the similar to previous one.

3.5 Lexical Similarity

There are many well-known orthographic metrics: edit distance, longest common subsequence ratio and rank distance. In this study we used longest common subsequence metric as implemented in python library. If a token cannot be normalized in all the steps above, the system takes the best matches from the vocabulary. If the similarity score is lower than a specific threshold k, 0.6 (empirically selected), candidate is rejected. If more than one close matches share the same score, the most frequent one is proposed as in the entire phases. If there is no any candidate, it proceeds the next phase. Those tokens that have not been handled by the system so far are passed to next n-gram language model where semantic similarity measurements are used for selection as defined in the next section.

3.6 Simplified N-gram Language Model

N-grams language model is considered still the state-of-the-art language model to measure semantic similarity. The language model is using only the prior probability of the word sequence. To extract contextually similar *(OOV, IV)* pairs from a large collection of micro-blog site, tokens are represented in context word vector. The cosine similarity is widely used to compare two context vectors. For a given ill-formed word, the context words are indexed with their positions within a context window size. Standard way of construction of contextual similarity is visiting each occurrence of ill-formed words to produce a vector within a window size. The candidate whose vector is the closest to that of ill-formed token is selected. The weak point of this approach is that producing context vector needs some re-occurrence of ill-formed word. Ill-formed words are more likely to be unique or low frequent.

Moreover, we simplify the approach to reduce the complexity, when considering the huge size of corpus and text messages. Thus, simply taking neighbors of ill-formed words and looking their context vectors could be easier and less complex. Even a given ill-formed word is unique, the system exploits only those nearby words. For instance, for a given case *(left, ill-token, right)*, we build a vector of words that occur with the *pattern (left, *, right)* as in the formula if and only if either left or right must be IV word.

$$arg\ max_c\ LexicalSim(c, token),\ \ c \in matchPattern(left, *, right) \tag{1}$$

The words captured by the pattern are counted and sorted by their pattern frequencies and lexical similarities. The main defect of the solution is that both nearby words are ill-formed.

4 Results and Evaluation

4.1 Evaluation

In this experiment, we build a broader IV list by exploiting dictionary[2] and a reliable corpus. This is more than a standard dictionary in that we need to keep all formation of the words in order to reduce the time complexity of parsing a word.The number of unique tokens in Turkish language is very high when compared to English and some other languages.The size is about ~ 2.2 million. For a given token, each step produces normalization candidates that are checked against that IV and if a match is found, the process is terminated.

Language usage in social media might vary due to culture, society, age, gender etc. When manually checking the system performance on Twitter data in Turkish Language with size of 1000 entries, we listed several problems and their distributions as shown in Table 1. In this table, two columns represents *model* and *model+n-gram*, where former refers to all steps until lexical similarity, latter is entire system including n-gram language model. The main problem originates from typing ASCII character instead of Turkish accents, with the ratio of 41%. Second problem is accent usage with the ratio of 16.4% due to socio-cultural, new generation etc. as discussed in previous sections. Third one is some typos such as dropping vowels, incorrectly typing some letters etc. Some suffixes -da -de must be separately added with a space. However, they are mostly typed adjacent so that the parser and the IV list cannot specify the word. Those are needed to be specified and solved before.

The system performance is listed in Table 1 for each subproblem where the baseline algorithm is simply choosing the lexically closest candidate from IV list. The table suggests that the model successfully solve asciification problem. Second, accent usage and repetition are also easily captured. While baseline function gets 40% precision, the proposed model gets about 77% precision and the n-gram supported model with gets 80%. Even though individual performance of n-gram language model is acceptable level, it improves system performance by only 3% increase due to that our initial model can capture many problems and remaining parts are limited and hard to be solved such that some unsolved cases could not be normalized by even human annotator.

Beside its contribution to system performance, it is worth testing how much n-gram model individually performs. We tested n-gram model against both the real twitter data set and artificially created data set. Success rate on real twitter set is 53 %. Artificial ill-formed words are automatically malformed by shifting, repeating, shuffling, deleting of original IV words in tweet data. Thus, we can measure the system performance against even huge amount of twitter data set,

[2] code.google.com/p/zemberek/

Table 1. Success rate of Models

	%	Baseline%	Model%	Model+N-gram%
Accent	**16.43**	38.04	75.00	75.00
ASCII	**41.25**	37.66	93.94	94.81
Slang	1.43	12.50	50.00	50.00
Proper	7.68	67.44	76.74	76.74
Vocatives	4.64	30.77	50.00	61.54
Type	**9.29**	51.92	61.54	67.31
Repetition	3.39	42.11	68.42	68.42
Unknown	6.07	0.00	0.00	0.00
No prob	3.75	28.57	52.38	66.67
Other	0.89	0.00	80.00	80.00
DeASCII	5.18	31.03	41.38	48.28
Average		0.40	**0.77**	**0.80**

as huge as possible. For artificially created over 16K ill-formed words, n-gram model showed a performance of 47% precision.

Artificial word formation is considered risky in other approaches, especially based on purely lexical based methods. Computer randomization might have some hidden rules even if randomization process is carefully and well prepared and that makes the model success invalid. Therefore our purely lexical-based models are checked against not the artificial data but the real data with real human mistakes. This is why we used two kinds of test set with different size. Therefore the size of real data set is less than that of artificial one. Running only n-gram model on artificial dataset does not have invalidity problem since that the model check neighbors that are not artificial and appears in real tweets. Only the target words are malformed. To clarify the situation, the average lexical similarity between artificially malformed words and their original ones is nearly 70% which is the similar average score in real data set. For example, the similarity between *"tmrrww"* and *"tomorrow"* is 71%, then it is acceptable level of artificial malformation whereas n-gram model applies lexical similarity only for sorting the candidates captured by distributional word space model.

The proposed model performs acceptable level of precision when compared to other studies. The results in [10] shows that 93% accuracy for English Tweets. Another proposed approach uses Random Walks on a contextual similarity graph [6] has 92.43% precision. The study [1] proposed a system that uses SMT with 79% accuracy. In Turkish [19], the proposed model with a 86% accuracy of ill formed word detection and 71% accuracy for candidate word generation.

In this paper, we propose an architecture of unsupervised text normalization Turkish language. The architecture orders the submodules by their degree of complexity. The important contribution is considered that lexical similarity functions, building a corpus-based IV dictionary and n-gram language modeling are integrated within an architecture for Turkish domain. For a better comparison with other languages, all the experiments and results need to be checked with same input data and the comparison criteria must be well-designed in term of time complexity, data characteristics, circumferences and other factors. That would be in our future plan. Other future work is to create a more reliable look-up dictionary because automatically augmented dictionaries can contains some level of incorrect entries. Reducing the those errors improves the system performance both in accuracy and complexity. Comparing the model with other studies and using BLEU system as baseline will be in our other future tasks as well.

5 Conclusion

In this paper, we propose an architecture for Turkish text normalisation. Even though Turkish is one of the less-studied language, Internet usage of native Turkish speakers is more than expected. Therefore the studies on social media texting language are getting important. They necessitates some preprocessing phases such as text normalization, tokenization, boundry identification etc.

We exploit many resources to normalize a given ill-formed token; tweet data, a huge corpus and a reliable dictionary. In order to design an architecture, first we figured out texting language phenomena in social media short text. The problems can vary; letter repetition, missing vowels, asciification and others. The submodules are sorted in accordance with their time complexity. We utilized lexical similarity to compare the candidates and IV words to make a decision. The last phase of the model is semantic model that includes n-gram language modeling. We simplified n-gram language model to reduce time complexity. This module uses word space model to normalize an ill-formed token by looking at their neighbors. Neighbors are used as semantic space that is very similary to lexico-syntactic patterns. The candidates that are captured from this phase are again evaluated in terms of their lexical similarity. So this step actually employs both semantic and lexical similarity.

We divide our architecture into two models; model and model+n-gram. The first model is based on only lexical similarity, which refers to all phases except n-gram. Second is using both lexical and semantic similarity, entire architecture including n-gram model. First model achieves 77 % precision, second model gets 80 % precision. Both highly outperform baseline function that simply takes the lexically closest term in IV to ill-formed token and its success rate is about 40 %. We also evaluated n-gram model separately to check its success. When we run n-gram semantic model as a separate unit on the data, we get 53% precision. When running on an artificially created big data, we get about 47%. This shows us designed n-gram language model can have a sufficient capacity even though its contribution to entire architecture is limited.

This study might be considered first study that applies both lexical similarity and semantic similarity together to Turkish text normalization. When comparing the similar studies, the proposed architecture gets sufficient and promising results. For a better comparison with those studies, all the experiments might be tested with same data and conditions. That would be in our future plan. Another future work is to extend look-up dictionary to reduce level of incorrect entries, which might improves the system performance. Using BLEU system for a better comparison will be in our other future tasks as well.

Acknowledgments

This paper was supported by Tübitak Program 1001 Scientific and Technological Research Council Projects Funding Program. Project (Code 113K175): Türkiye'de Kamuoyunun Bilime Yönelik Tutum Değişiminin Ölçülmesinde Metin Madenciliği Yolu ile Alternatif Bir Sistem

References

1. Max Kaufmann and Jugal Kalita. Syntactic normalization of Twitter messages. In *Proc. of the 8th International Conference on Natural Language Processing (ICON 2010), Kharagpur, India*, (2010)
2. Deana Pennell and Yang Liu. A character-level machine translation approach for normalization of sms abbreviations. In *Proc. of 5th International Joint Conference on Natural Language Processing*, pages 974–982 (2011)
3. Aiti Aw, Min Zhang, Juan Xiao and Jian Su. A phrase-based statistical model for sms text normalization. In *Proc. of the COLING/ACL on Main conference poster sessions (COLING-ACL '06). Association for Computational Linguistics, Stroudsburg, PA, USA*, pages 33–40 (2006)
4. Richard Beaufort, Sophie Roekhaut, Louise-Amelie Cougnon and Cedrick Fairon. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10). Association for Computational Linguistics, Stroudsburg, PA, USA*, pages 770–779 (2010)
5. Paul Cook and Suzanne Stevenson. Unsupervised model for text message normalization. In *Proc. of the Workshop on Computational Approaches to Linguistic Creativity , CALC '09, Association for Computational Linguistics, Stroudsburg, PA, USA*, pages 71–78 (2009)
6. Hany Hassan and Arul Menezes. Social text normalization using contextual graph random walks. In *Proc. of the 51st ACL, Association for Computational Linguistics, Sofia, Bulgaria*, pages 1577–1586 (2013)
7. Fei Liu, Fuliang Weng and Xiao Jiang. A broad-coverage normalization system for social media language. In *Proc. of the 50th ACL, Association for Computational Linguistics, Stroudsburg, PA, USA*, pages 1035–1044 (2012)
8. Clark and Kenji Araki. Text normalization in social media: progress, problems and applications for a pre-processing system of casual english. *Procedia-Social and Behavioral Sciences*, 27:2–11 (2011)
9. Congle Zhang, Tyler Baldwin, Howard Ho, Benny Kimelfeld and Yunyao Li. Adaptive parser-centric text normalization. In *Proc. of the 51st ACL, Association for Computational Linguistics, Sofia, Bulgaria*, pages 1159–1168 (2013)
10. Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proc. of the 49th ACL HLT, Association for Computational Linguistics, Portland, Oregon, USA*, pages 368–378 (2011)

11. Bo Han, Paul Cook and Timothy Baldwin. Automatically Constructing a Normalisation Dictionary for Microblogs. In *EMNLP-CoNLL 2012, Jeju, Republic of Korea*, pages 421–432 (2012)
12. Bo Han, Paul Cook and Timothy Baldwin. Lexical Normalisation of Short Text Messages. In *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1) 5:1–5:27 (2013)
13. Bo Han, Paul Cook and Timothy Baldwin. Spanish Text Normalisation. In *Proc. of Tweet-norm: Tweet Normalization Workshop at SEPLN 2013, 67-71, Madrid, Spain*, pages 67–71 (2013)
14. Uladzimir Sidarenka, Tatjana Scheer and Manfred Stede. Rule-Based Normalization of German Twitter Messages. In *Proc. of the GSCL Workshop Verarbeitung und Annotation von Sprachdaten aus Genres internetbasierter Kommunikation*, (2013)
15. Catherine Kobus, François Yvon and Geraldine Damnati. Normalizing SMS: are two metaphors better than one?. In *Proc. of the 22nd International Conference on Computational Linguistics - Volume 1 (COLING '08), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA*, pages 441–448 (2008)
16. Pablo Ruiz, Montse Cuadros and Thierry Etchegoyhen. Lexical Normalization of Spanish Tweets with Rule-Based Components and Language Models. *Procesamiento del Lenguaje Natural*, 52:45–52 (2014)
17. Jordi Porta and Jose Luis Sancho. Word normalization in Twitter using finite-state transducers. In *Workshop on Tweet Normalization at SEPLN Tweet-Norm*, (2013)
18. Mohammad Arshi Saloot, Norisma Idris and Rohana Mahmud. An architecture for Malay Tweet normalization. *Information Processing and Management*, 50(5):621–633 (2014)
19. Dilara Torunoğlu and Gülşen Eryiğit. A Cascaded Approach for Social Media Text Normalization of Turkish. In *5th Workshop on Language Analysis for Social Media (LASM) at EACL, Gothenburg, Sweden*, (2014)
20. Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. ACM Trans. Speech Lang. Process., 5(1) 3:1–3:29 (2007)

Design Issues in Automatic Grapheme-to-Phoneme Conversion for Standard *Yorùbá*

Abímbọ́lá R. Ìyàndá and Ọdẹ́túnjí A. Ọdẹ́jọbí

Obafemi Awolowo University
Computer Science and Engineering Department
Ile-Ife, Nigeria
abiyanda@oauife.edu.ng, oodejobi@oauife.edu.ng

Abstract. Grapheme-to-Phoneme (G2P) conversion is an important problem in Human Language Processing development, particularly Text-to-Speech (TTS). Its primary goal is to accurately compute the pronunciation of words in the input texts. This work examines design issues with respect to components of the automatic G2P for standard *Yorùbá* (SY). The automatic process includes: (i) Tokenisation of Input, (ii) Identification of nasal characters, (iii) Syllabification, and (iv) Conversion of Graphemes to Phonemes. The structure of the Yoruba text is described and a text corpus design for standard *Yorùbá* TTS is presented. The analysis of the data was done using Zipf's law. The outcome of this work provided adequate requirement for the design.

1 Introduction

Grapheme-to-phoneme conversion (G2P) refers to the task of finding the pronunciation of a word given its written form. It has important applications in human language technologies

Alphabetic writing systems are based on the idea that the orthographic form is a conventional representation of a word's pronunciation [1]. In a perfectly phonological alphabet such as existed in *Yorùbá* language, there is a one-to-one correspondence between letters (graphemes) and phonemes.

The Grapheme-to-Phoneme (G2P) conversion subsystem is a crucial component of a TTS particularly for a tone language. TTS is an important human language technology which requires a good G2P conversion process. It is impossible to have a good-working text-to-speech system without having a tool that generates correct pronunciation when presented with the word orthography.

2 Standard Yoruba language

Yorùbá language is one of the three major indigenous languages, along with Hausa and Igbo in Nigeria and it is spoken by over 37 million people [2]. *Yorùbá* is a native language of the *Yorùbá* people, an ethnic group primarily located in

pp. 195–205; rec. 2015-01-29; acc. 2015-02-25

south-western Nigeria (Lagos, *Ọ̀yọ́, Ògùn, Oǹdó, Èkìtì, Ọ̀ṣun* and parts of *Kwara* and *Kogí* states). SY is used in language education, the mass media and everyday communication [3]. *Yorùbá* is a tone language in which a pitch of an utterance is used to express differences in meaning; or when a particular pitch or change of pitch constitute an element in the intonation of a phrase or sentence, such as high, mid or low. The tone structure in *Yorùbá* is made possible by diacritics - tone marking on top of vowels and syllabic nasals to give meaning to the contexts.

2.1 Description of *Yorùbá* Text

Yorùbá alphabet comprises of 18 consonants (represented graphemically by *b, d, f, g, gb, h, j, k, l, m, n, p, r, s, ṣ, t, w, y*) and 7 vowels represented graphemically by *a, e, ẹ, i, o, ọ, u* [4]. It should be noted that the *gb* is a diagraph i.e. a consonant written with two letters. Five nasal vowels exist in the language by adding *n* with the oral vowels (*a, e, ẹ, i, o, ọ, u*) which is represented graphemically as *an, ẹn, in, ọn, un*. Also, one syllabic nasals represented graphemically as *n* exists. It should also be noted that *an* and *ọn* are phonemically the same. The consonant and vowel systems as well as a more detailed description of SY are presented in [5].

In SY language, syllables are considered as the basic unit of sound because it has been established as a perceptually and acoustically coherent unit [4]. In addition, syllable can be considered as the basic unit of G2P in tone language as agreed with the work of [6], [7] and [8] established that syllables produce reasonably natural quality speech. The letters are combined to form syllable based on the syllable structures and syllables combined to form words. Words are combined to form statements. There exist five syllables structures in the language and these are: V (oral vowels), Vn (nasal vowels), N (syllabic nasal), CV (combination of consonants and oral vowels) as well as CVn (combination of consonants and nasal vowels) [3]

The syllables are the tone bearing elements of the SY language. For example, in the statement: *Abím̄bọ́lá ti lọ sí oko* (*Abím̄bọ́lá* has gone to the farm), there are ten syllables. This also implies that there exist ten tones (MHMHH-M-M-H-MM).

3 Data

The domain for the speech synthesis for this research is in language education, religious and mass media from various sources such as Internet, digitized printed material and existing digital materials from non-Internet sources. Two SY newspaper (Aláròyé and *Yorùbá* Gbòde) and two SY textbooks ([9], [10]) were selected. The two newspapers were not toned mark, and *Yorùbá* Gbòde is without under dots. The texts were edited using Tákàdá text editor (www.sourceforge.net/projects/takada) and were corrected for graphemic items (tone marks and under dots) using Àkọtọ́ *Yorùbá* [11]. Sample data is as shown in Table 1. To increase the quality and coverage of the acquired materials, additional *Yorùbá* text data were gathered from existing printed materials by scanning through a process known as optical character recognition. The volume of textual data generated using these techniques was about 291,392 words.

Table 1. Sample Data

Yoruba Statement	Meaning
Ìbùkún ni fún ọkùnrin náà tí kò rìn ní ìmọ̀ àwọn ènìyàn búburú, tí kò dúró ní ọ̀nà àwọn ẹlẹ́ṣẹ̀, àti tí kò sì jókòó ní ibùjókòó àwọn ẹlẹ́gàn. Ṣùgbọ́n dídùn inúu rẹ̀ wà nínú òfin Olúwa, àti nínú òfin náà ni ó ńṣe àsàrò ní ọ̀sán àti ní òru. Yóò sì dàbí igi tí a gbìn sí etí ipa odò, tí ó ń so èso rẹ̀ jáde ní àkókò rẹ̀, ewée rẹ̀ kì yóò sì rẹ̀, àti ohunkóhun tí ó bá ńṣe ni yóò máa ṣe déédé. Nítorí Olúwa mọ ọ̀nà àwọn olódodo, ṣùgbọ́n ọ̀nà àwọn ènìyàn búburú ni yóò ṣègbé.	Blessed is the man that walks not in the counsel of the ungodly, nor stands in the way of sinners, nor sits in the seat of the scornful, but his delight is in the law of the LORD; and in his law he does meditates day and night. He shall be like a tree planted by the rivers of water, that brings forth its fruit in its season; its leaf also shall not wither; and whatsoever it does shall prosper. For the LORD knows the way of the righteous: but the way of the ungodly shall perish.

Data analysis was done using Zipf's law, which observed a phenomenon in human languages (and some other social structures) that indicated that there is a pattern in the distribution of tokens and that a few very frequent words make up a very large portion of any text or collection of texts, while the large majority of words occur relatively rarely. Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc [12].

The analysis of the research data for frequency distribution was performed using a freeware concordance; Simple Text Analysis Tool (TextSTAT) version 2.9.0.0 [13]. This tool was used to generate the frequency count for the word tokens in the text data. It was also used to detect words that have incorrect forms such as wrong diacritics (tone marks and under dots). In the analysis of the words in the text corpus, 6857 words appeared only once in the corpus and the word with the highest appearance in the data set *(tí)* occur 8257 times, followed by *ni* with frequency of 6358 and followed by *àwon* with frequency of 5753. This pattern is expected of natural language [12]. Table 2 shows the distribution of some selected words in conformity with Zipf's law for linguistic data.

Zipf's law states that there is an inverse proportional relationship between the rank and frequency of words in a text. Rank is the position of a word in

Table 2. Ranks and frequencies of the text data

Word type	Rank (r)	Frequency (f)	Proportion (%)
n	10	4365	1.4978
ti	20	2673	0.9173
fún (give)	30	2215	0.7601
gẹ (cut)	40	1560	0.5354
èdè (language)	50	1153	0.3957
o (you)	60	947	0.3250
ìṣe (doing)	70	786	0.2697
ìtàn (story)	80	694	0.2382
pọ	90	599	0.2056
ìmọ̀ (knowledge)	100	524	0.1798
ìpínlẹ̀ (state)	200	189	0.0649
àtijọ́ (former)	300	110	0.0377
yí (turn)	400	77	0.0264
ọjà (market)	500	57	0.0196
Adéyẹmí (*Yorùbá* name)	600	44	0.0151
ààyè (opportunity)	700	37	0.0127
àgbàlagbà (adult)	800	30	0.0103
gbeyẹ̀wò (consider)	900	25	0.0086
déésì (translated name)	1000	21	0.0072
kórìíra (to hate)	10000	1	0.0003

a table of words ordered by frequency of occurrence (rank one being the most frequent) [12].

The general nature of Zipf's distribution as it applies to human language use whether in oral or written communication or discourse is as highlighted below [11]:

i. a few word tends to score very high on the frequency table meaning that they appear regularly in discourse;
ii. a medium number of words or linguistic tokens appears with relatively medium frequency indicating that they are regularly used;
iii. a huge number of words in a document have very low level of occurrence, indicating that they are rarely used.

With the proportion of these tokens in relation to the total token count, it means that these words occurred in many contexts and are good features to be used in grapheme to phoneme conversion system for *Yorùbá* language. The Zipf's curve for the words whose rank ranges from tens to thousands (the regularly occur to the rarely occur), to have a wide coverage of the database is shown in Figure 1. The shape of the curve justifies Zipf's law for linguistic data. This means that a few word appear regularly in discourse; a medium number of words or linguistic tokens appears with relatively medium frequency indicating that they are averagely used; a huge number of words in a document have very low level of occurrence, indicating that they are rarely used.

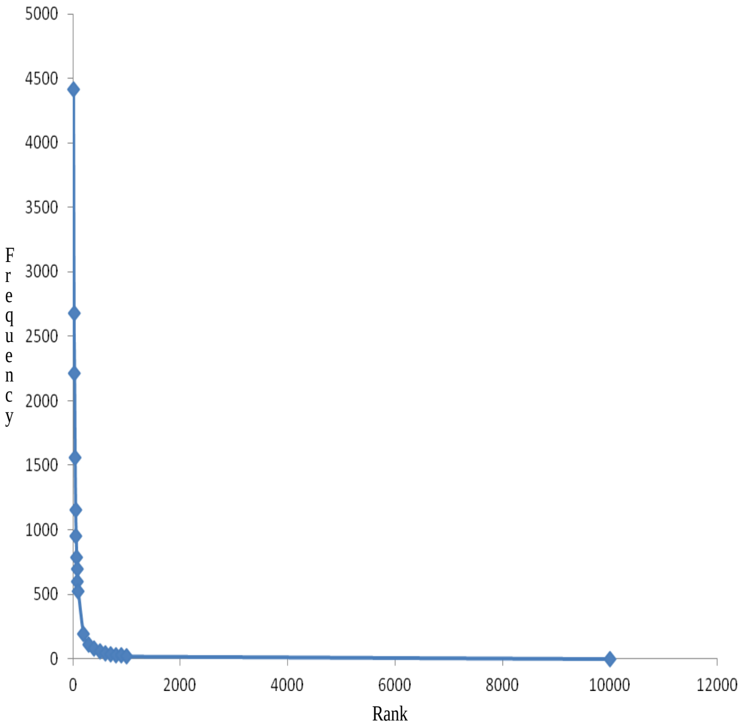


Fig. 1. Zipf frequency-rank curve for *Yorùbá* texts

4 Processes in G2P Conversion

The processes involved in the G2P conversion for SY is shown in Figure 2. These processes include: (i) Input tokenisation process which splits the input sentence into individual characters, (ii) Nasal characters identification process which detects nasalised vowels, (iii) Syllabification process which accepts input tokens and produces corresponding syllables, and (iv) Graphemes to Phonemes conversion process. To facilitate this process of converting grapheme to phoneme, the characters with underdots and the diagraph- *ẹ̀, ẹ́, ọ̀, ọ́* and *gb* were transformed to 'z', 'x', 'c', 'v', and 'q' respectively.

4.1 Input Tokenisation

The input to this process is a text file. Tokenisation takes the input text and breaks it into sentences marked by a new-line, and each sentence is further breakdown into chunks called tokens. Tokenisation was done using white spaces as delimiters. This process was achieved using the *split()* function of the String class. This function takes a string and separates them using white space. The

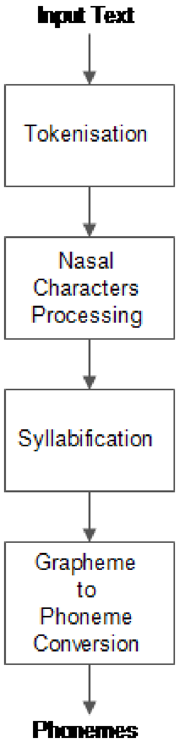


Fig. 2. G2P Processes

output of the *split()* function is a List object containing all the tokens. Each token is separated by a front slash (/).

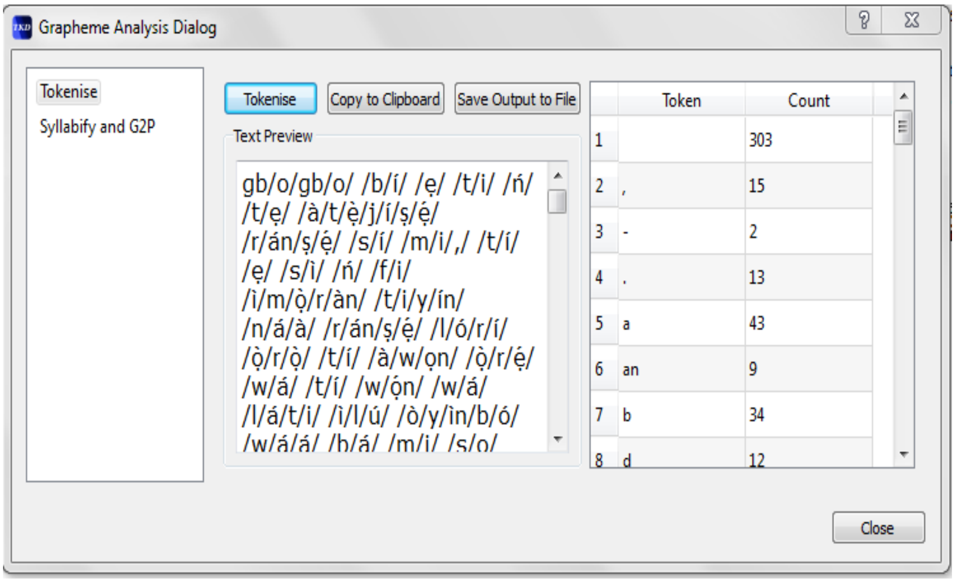
Figure 3 shows the output for Software processing of tokenisation.

4.2 Process Nasal Characters

This stage processes nasal characters. For example: to convert *’a’* and *’n’* to *’an’*; *’ẹ’* and *’n’* to *’ẹn’*; *’i’* and *’n’* to *’in’*; *’ọ’* and *’n’* to *’ọn’*; and *’u’* and *’n’* to *’un’* where appropriate.

It is should be noted that any of the consonant can precede nasal vowel except l,m and n. E.g.*imún*, *inán* as well as *lan, lẹn, lọn, lun* does not exist orthographically in SY words. The process is achieved by the following lines of code.

```
def processNasals(self, sym=[]):
        ind = [i for i, x in enumerate(sym) if x == 'n']
        for i in ind:
            if i != 0:
```



gb/o/gb/o/ /b/í/ /ẹ/ /t/i/ /ń/
/t/ẹ/ /à/t/ẹ̀/j/í/ṣ/ẹ́/
/r/án/ṣ/ẹ́/ /s/í/ /m/i/,/ /t/í/
/ẹ/ /s/ì/ /ń/ /f/i/
/ì/m/ọ̀/r/àn/ /t/i/y/ín/
/n/á/à/ /r/án/ṣ/ẹ́/ /l/ó/r/í/
/ọ̀/r/ọ̀/ /t/í/ /à/w/ọn/ /ọ̀/r/ẹ́/
/w/á/ /t/í/ /w/ọ́n/ /w/á/
/l/á/t/i/ /ì/l/ú/ /ò/y/ìn/b/ó/
/w/á/á/ /b/á/ /m/i/ /s/o/

	Token	Count
1		303
2	,	15
3	-	2
4	.	13
5	a	43
6	an	9
7	b	34
8	d	12

Fig. 3. Screen shot for sentence tokenised

```
        vn = sym[i - 1] + 'n'
        if vn in self.Vn:
            if i + 1 < len(sym):
                if sym[i + 1] not in self.V:
                    sym[i - 1] = vn
                    sym[i] = u'@'
            else:
                sym[i - 1] = vn
                sym[i] = u'@'

# remove all x's
for i, x in enumerate(sym):
    if x == '@':
        sym.remove('@')
return sym
```

4.3 Syllabification Process

The syllable corpus was created from the word corpus using a process known as syllabification. The syllabification was achieved using a Push Down Transducer (PDT) [14]. The PDT employed for the syllabification process extends the functionality of PDA (generalised NFA) beyond language recognition by ensuring an output on every transition. This is necessary because we are not interested in verifying the acceptance of the input, but rather to obtain *Yorùbá* syllables from a continuous stream of texts. As a result, the PDT developed is a 8-tuple defined as:

$$PDT =< Q, \varSigma_i, \varSigma_o, \varGamma, q_0, Z_0, F, \delta > \tag{1}$$

Where:

- Q is a finite set of states, and
  $Q = \{Q_0, Q_1, Q_2\}$
- $\varSigma_i$ is a finite set of input symbols (tokens), and
  $\varSigma_i = V \cup V_n \cup C \cup N$, where
  V is a set of vowels, i.e.,
  $V = \{$ *a, e, ẹ, i, o, ọ, u, à, á, è, é, ẹ̀, ẹ́, ì, í, ò, ó, ọ̀, ọ́, ù, ú* $\}$
  V_n is a set of nasalised vowels, i.e.,
  $V_n = \{$ *an, ẹn, in, ọn, un, àn, án, ẹ̀n, ẹ́n, ìn, ín, ọ̀n, ọ́n, ùn, ún* $\}$
  C is a set of consonants, i.e., $C = \{$*b, d, f, g, gb, h, j, k, l, p, r, s, ṣ, t, w, y*$\}$
  N is a set of syllabic nasals, i.e., $N = \{$*m, n, ń, ǹ, ḿ*$\}$
- $\varSigma_o$ is a finite set of output syllable symbols. This represents the acceptable sequence of characters that make up a syllable. As such,
  $\varSigma_o = V, v_n, CV, CV_n, N$
- $\varGamma$ is a finite set of stack symbols, i.e., $\varGamma = C \cup N \cup \{\lambda\}$
- $q_0 \in Q$ is an initial state, and $q_0 = Q_0$
- $Z_0 \in \varGamma$ is the initial stack symbol, and $Z_0 = \lambda$
- $F \subseteq Q$ is a set of final states, and $F = Q_3$
- δ is a transition function given as:

$$Q \times \varSigma_i \times (\varGamma \cup \epsilon) \to 2^{Q \times \varGamma \times \varSigma_o} \tag{2}$$

A transition of a PDT could be represented using an element $(p, a, A, q, B, b) \in \delta$ which is represented as shown in Figure 4 is interpreted as: In a state $p \in Q$, given an input symbol $a \in (\varSigma_i \cup \{\epsilon\})$, and $A \in \varGamma$ being the topmost stack symbol, move to state $q \in Q$, pop $A \in \varGamma$ and push $B \in \varGamma$ onto the stack , and produce an output symbol $b \in (\varSigma_o \cup \{\epsilon\})$.

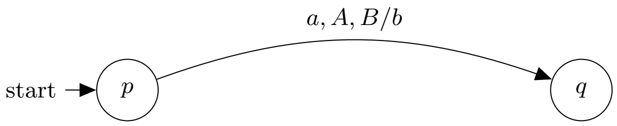


Fig. 4. A transition $(p, a, A, q, B, b) \in \delta$ of a PDT

As found in PDAs, PDTs have two modes of acceptance which are: acceptance by final state and acceptance by empty stack. We employed the two modes of acceptance in the modified PDA such that the PDA accepts the sequence of symbols w if there is a sequence of transitions from $(q_0, a_i, \lambda, q_i, B_i, b)$ and terminates at $(q_i, \#, \lambda, \epsilon, q_F \in F, \epsilon, \epsilon)$. This means that for a sequence of input

Algorithm 1: Algorithm for the Nasal Vowel Identification and Syllabification Processes.

```
V = {a, e, ẹ, i, o, ọ, u, á, à, é, è, ẹ́, ẹ̀, í, ì, ó, ò, ọ́, ọ̀, ú, ù } ;
C = {b, d, f, g, gb, h, j, k, l, p, r, s, ṣ, t, w, y} ;
$V_n$ = {an, ẹn, in, ọn, un, àn, án, ẹ̀n, ẹ́n, ìn, ín, ọ̀n, ọ́n, ùn, ún } ;
N = {m, n, ń, ǹ, ḿ} ;
Data: tokenList: A List of Input Tokens
Result: syllables: A List of Syllables
procedure syllabificator (tokenList)
    for token in enumerate(tokens) do
        syllables = ";
        stack = new Stack();
        if $token \in (V \cup Vn)$ then
            if size(stack) == 0 then
                syllables =+ token;
            else
                top = stack.pop();
                if $top \in (self.C \cup self.N)$ then
                    syllables =+ top + token;
                end if
            end if
        else if $token \in (self.C \cup self.N)$ then
            if size(stack) == 0 then
                stack.push(token);
            else
                top = stack.pop();
                if $top \in self.N$ then
                    stack.push(token);
                    syllables =+ top;
                end if
            end if
        else
            syllables =+ token;
        end if
        return syllables ;
    end
end procedure
```

symbols to be accepted the PDT must be in a final state, the stack must be empty, and all the input symbols must be read. The PDT was implemented using the algorithm presented in Algorithm 1.

The PDT was simulated in JFLAP using the following input tokens:

tokens = [‘ǹ’, ‘ǹ’, ‘k’, ‘an’, ‘ ’, ‘ḿ’, ‘b’, ‘ẹ’, ‘ ’, ‘l’, ‘á’, ‘b’, ‘ẹ́”, ‘ ’, ‘ọ̀’, ‘ọ̀’, ‘r’, ‘un]

The transition table for simulation of the syllabification process using these input tokens is shown in Table 3, and the output of the simulation is *'n-n-kan m-be la-be o-o-run'*

Table 3. Syllabification Process for *'ǹǹkan ḿbẹ lábẹ́ ọ̀ọ̀run#'*

Token	Current	Current	Pop	Push	Resulting	Output	Next State
	State	Stack			Stack		
'ǹ'	Q_0	[λ]	λ	ǹ, λ	[ǹ, λ]	ϵ	Q_2
'ǹ'	Q_2	[ǹ, λ]	ǹ	ǹ	[ǹ, λ]	ǹ	Q_2
'k'	Q_2	[ǹ, λ]	ǹ	k	[k, λ]	ǹ	Q_2
'an'	Q_2	[k, λ]	k	ϵ	[λ]	kan	Q_1
' '	Q_1	[λ]	λ	λ	[λ]	' '	Q_1
'ḿ'	Q_1	[λ]	λ	ḿ,λ	[m, λ]	ϵ	Q_2
'b'	Q_2	[m, λ]	ḿ	b	[b, λ]	ḿ	Q_2
'ẹ'	Q_2	[b, λ]	b	ϵ	[λ]	bẹ	Q_1
' '	Q_1	[λ]	λ	λ	[λ]	' '	Q_1
l	Q_1	[λ]	λ	l, λ	[l, λ]	ϵ	Q_2
á	Q_2	[l, λ]	l	ϵ	[λ]	lá	Q_1
b	Q_1	[λ]	λ	b, λ	[b, λ]	ϵ	Q_2
ẹ́	Q_2	[b, λ]	b	ϵ	[λ]	bẹ́	Q_1
' '	Q_1	[λ]	λ	λ	[λ]	' '	Q_1
ọ̀	Q_1	[λ]	λ	λ	[λ]	ọ̀	Q_1
ọ̀	Q_1	[λ]	λ	λ	[λ]	ọ̀	Q_1
r	Q_1	[λ]	λ	a, λ	[r, λ]	ϵ	Q_2
un	Q_2	[r, λ]	r	ϵ	[λ]	run	Q_1
#	Q_2	[λ]	λ	ϵ	[]	ϵ	$Q_3 \in F$

4.4 Graphemes to Phonemes conversion

This stage converts all the graphemes resulting from the previous stage into phonemes. Different methods for building automatic alignments have been proposed, among which there are one-to-one and many-to-many alignments [1]. In this study one-to-one alignment was adopted because each letter corresponds only to one phoneme and vice versa. This is achieved by first replacing all transformed characters back to their original characters. For example, grapheme ẹ which was formally replaced with 'z' was converted back to ẹ before the phonemic transcription output was given. Syllables with high, mid and low tones are transcribed with their tones (H, M, L, respectively) placed in bracket beside them.

5 Conclusion

The Grapheme-to-Phoneme (G2P) conversion subsystem is a crucial component of a TTS particularly for a tone language. Attempts have been made to develop other components of Standard *Yorùbá* TTS, but the G2P subsystem is yet to be addressed.

This work opens the way for synthetic speech to approach the quality of natural speech. In this paper we outlined the fundamentals of this concept and gave a detailed progress report on our ongoing research. We gave an account of the data analysis performed and the process underlying SY text to sound was examined. Some other design issues were as well discussed . This work provides adequate requirement for the design and as well provides the basis for research and development in TTS for SY language.

References

1. Bisani, M., Ney, H.: Joint-Sequence Models for Grapheme-to-Phoneme Conversion. Speech Communication **50** (2008) 434–451
2. CIA: Central intelligence agency world factbook 2014. Available online at http://www.cia.gov/cia/publications/factbook. (2014) Visited April, 2014.
3. Odéjobí, O.A.: A quantitative model of yorùbá speech intonation using stem-ml. INFOCOMP Journal of Computer Science **6** (2007) 47–55
4. Adewole, L.O.: The Categorical Status and Function of the Yoruba Auxiliary Verb with some Structural Analysis in GPSG. PhD thesis, University of Edinburgh, Edinburgh (1988)
5. Iyanda, A.R.: Design and Implementation of a Grapheme-to-Phoneme Conversion System for Yorùbá Text-to-Speech Synthesis. PhD thesis, Obafemi Awolowo University, Ile-Ife, Nigeria (2014)
6. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics **23** (1997) 269–311
7. Gakuru, M., Iraki, F.K., Tucker, R.C.F., Shalonova, K., Ngugi, K.: Development of a kiswahili text to speech system. In: INTERSPEECH, ISCA (2005) 1481–1484
8. Anberbir, T., Takara, T.: Development of an amharic text-to-speech system using cepstral method. Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages AfLaT, Athens, Greece ©Association for Computational Linguistics (2009) 46–52
9. Awobùlúyì, O.: Ẹ̀kọ́ Ìṣẹ̀dá-Ọ̀rọ̀ *Yorùbá* . Montem Paperbacks, Akure, Ondo State (2008)
10. Owólabí, K.: Ìjìnlẹ̀ Ìtúpalẹ̀ Èdè Yorùbá (1): Fònẹ́tîkì ati Fonọ́lọjì. Universal Akada Books Nigeria Limited (2011)
11. Asahiah, F.O.: The development of a Standard *Yorùbá* Diacritics Restoration System. PhD thesis, Obafemi Awolwowo University, Ile-Ife, Nigeria (2014)
12. Sorell, C.: Zipf's law and vocabulary. (2012) In Carol A. Chapelle, editor, The Encyclopedia of Applied Linguistics. Wiley, New York, NY, USA. ISBN 1-4051-9843-5.
13. Hüning, M.: Textstat - simple text analysis tool/ concordance software. available from http://neon.niederlandistik.fu-berlin.de/en/textstat/. visited: April, 2014. (2014)
14. Choffrut, C., Culik-II, K.: Properties of finite and pushdown transducers. SIAM J. Comput. **12** (1983) 300–315

Using Sentiment Analysis to Detect Customer Attitudes in Social Media Comments

Ahmet Süerdem[1,],Eda Kaya[1]

[1]İstanbul Bilgi University, İstanbul, Turkey
asuerdem@bilgi.edu.tr, edakaya@gmail.com

Abstract .ig data analytics is increasingly replacing or complementing classical data collection methods like surveys. Business intelligence tools and systems now play a key role in decision-making and find applications in the areas such as customer profiling, customer support, market research, segmentation, brand monitoring. Transforming unstructured text data for structured quantitative analysis is an important challenge for business analytics. Sentiment analysis is a promising method in terms of transforming words into numbers to detect the tone of subjective expressions. The aim of this paper is scrutinize the role of sentiment analysis for emulating classical attitude and market research tools. We used sentiment analysis on product reviews data occurring in the e-market place in Turkey. We have applied a variety of machine learning algorithms and some term selection algorithms. Unigram approach for the term selection and Naive Bayes approach for machine learning have performed better than others. Our results suggest that sentiment analysis can be applicable for Turkish language after a rigorous text preprocessing and term selection process.

Keywords: Business Intelligence • Sentiment Analysis • Text Classification • Machine Learning

1 Introduction

Social media sites provide opportunities for gathering vast amounts of marketing intelligence information from naturally occurring data such as comments and evaluation about brands, individuals and products. Contrary to the surveys where attitude data can be collected as a one-off occasion, social media users share their opinions on a continuous basis in a naturally occurring environment. This provides management decision makers with instantly updated information and opportunity to collect data from many people which would physically impossible to do so. Although decision makers increasingly gather information from social media about the issues such as what consumers think about a new advertisement campaign, customer complaints and how consumers compare their brand to their competitors, systematic analysis of this information is still an emerging issue. Systematic analysis can establish the links between this information and corporate knowledge management systems.

The high cost of media monitoring of vast amounts of texts through human processing brings forth a need for data mining tools that can automatically analyze texts.

pp. 207–215; rec. 2015-02-16; acc. 2015-02-27

Accordingly, business intelligence and text analytics tools increasingly complement classical data collection instruments such as surveys and polls. These tools find growing application in the fields such as customer profiling, customer support, marketing research, market segmentation, and brand monitoring. However, although social media applications present a rich information source for data mining, data in this form is usually available as unstructured text. In this respect, Natural Language Processing and text mining techniques offer opportunities for structuring those data. The aim of this paper is to emulate conventional attitude and market research tools by using these techniques in the Turkish language context. For this purpose, we made machine learning based sentiment analysis of Facebook comments about two competing retail brands a priori categorized as positive or negative by the customers.

2 Theoretical Background

Automatic text analysis finds its grounds in classical content analysis [1]. However, this method usually focuses on determining factual attributes of the content by ignoring the evaluative elements in the language. Subjective aspects such as opinions, attitudes, evaluations, styles, etc. are essential for obtaining the gist of a text [2]. Accordingly, sentiment analysis appeared as a promising alternative in recent years to complement content analysis for detecting the evaluative tone of subjective expressions [3, 4]. Using text mining methods such as statistical natural language processors, part-of-speech tags, parsers and advanced lexicons, sentiment analysis claims to automatically detect whether a text contains a positive or negative tone [5]. Although sentiment analysis has started with the analysis of movie reviews [5, 6, 7, 8, 9], it now extends its domain to a variety of applications. Reviews and discussions on cars, banks, travel resorts [10], music, books, mobile phones, kitchenware, hotels [11], cameras, printers and baby strollers [12] are amongst the various topics that sentiment analysts are interested in.

Habitual method in sentiment analysis concerns developing automatic classifiers with supervised machine learning procedures applied to human annotated texts. This method is preferred because of its compelling performance for determining positive or negative tones in texts without encountering the constraints of linguistic analysis [5]. A notable alternative to this method is building semantic thesauri, which are based on processes like the counting of keywords expressing emotional polarity in a text [10]. Between these two methods, machine learning is preferred to the latter because it also covers the relevant context around the words [3, 13]. While keyword method can detect more precisely whether the chosen aspects belong to the predicted class (Precision: true positive rate), machine learning is strong in terms of retrieving the relevant instances (Recall: false negative rate) [14].

In supervised learning, the learning algorithm is trained by texts whose classes are annotated by humans. The algorithm derives a model from the texts selected as a training set through the relationship between input features (terms) and their class assignments. Features are weighted to the extent they demarcate between classes. The model is cross-validated through a test set not covering the text units in the training set. The overlap between actual and model predicted classes is accepted as the accuracy criteria [15].

While machine learning is commonly used in English, in other languages keyword method is the dominant approach. For example, Mihalcea [16] conducted sentiment analysis by translating mood state words from English to Romanian; Mathieu [17] to French; Cho and Lee [18] to Korean, and Abbasi [19] to Arabic. The limited success of the keyword approach turned the researchers of non-English languages to machine learning. For example, Kanamaru [20] has developed a machine learning algorithm for determining emotional states for Japanese language, while Xu [21] did the same for the Chinese. Pang [5] remarked that success of automatic classification for the languages other than English depends on the feature type selection such as unigrams or n-grams as well as the selection of best classifier algorithms. In this vein, this paper aims to contribute to the literature by experimenting with different machine learning algorithms on Turkish texts and testing the effect of different feature types (unigrams and n-gram) on the classification accuracy.

Sentiment analysis in Turkish is rare except a few cases (i.e. Boynukalın [22] classified Turkish texts according to emotional state expressions through machine learning; Eryiğit [23] compared machine learning with dictionary based methods). They found that feature selection process is the most critical decision for machine learning in Turkish language. Performance of feature selection depends on text pre-processing procedures such as stemming and lemmatizing, detection of synonyms, exclusion of function words. However this is a complex procedure for morphologically rich languages such as Arabic and Turkish, which requires special processes [19]. Moreover, unconventional language use in the social media context makes this issue more complex. Hence, in this paper we propose different algorithms to solve some of these problems. Our method consists of three steps: text pre-processing, feature selection and modeling.

3 Method

3.1 Text preprocessing

On preprocessing phase, we have particularly focused on the symbol use in the social media. For example, emoticons like smileys express information on user's emotions. To include them as features, we converted facial expressions (emoticons) into meaningful words such as:

:-) positive; :=) positive; :D very positive ; <3 love

Besides emoticons, unconventional abbreviations are frequently used in social media. We converted them into their synonyms for the algorithm not to detect them as different features:

ltf: please (lütfen); pls: please (lütfen); tşk: thanks (teşekkürler); inş: hopefully (inşallah); abv: damn (allah belanı versin).

Another unique social-media-only expression that defines the strength of sentiment is repeating characters. For example, nooooo is stronger than no.

The next pre-processing step involves spell checks. For this purpose we have used "Zemberek", an open source Turkish NLP library (https://github.com/ahmetaa/zemberek-nlp) which has a spelling dictionary with more than one million words. The example output presents original word and several suggestions (Table 1).

Table 1. The representative output for Zemberek's results

originalWord	**suggested1**	**suggested2**	**suggested3**
Tartisilir	tartışılır	Tartışılır	
Ihtiyacimiz	ihtiyacımız		
Paylasınca	paylaşınca	Paylaşınca	
Seninle	Seninle	Şeninle	şeninle
Arkadaslarn			

In the "originalWord" area the actual form of the word is seen. The fields that begin with "suggested.." label, show alternative suggestions for the word.

Hence, during the preprocessing phase we have converted frequently used emoticons and abbreviations into words, corrected misspellings and edited the repeating characters.

3.2 Feature selection

This step involves the generation and selection of features to be used in the classification algorithms. We compared two different approaches: i) "Bag of words": features are generated disregarding syntax and word sequences (unigrams) for keeping the variability in the text. ii) N-gram: takes into account word sequence patterns (syntax) and phrases. For each syntactic pattern and phrase, it generates a separate feature. Social media texts are usually short and do not include many N-grams. While N-grams have been found to perform better for longer texts, they can discount some semantic information in shorter texts in accordance with the Zipf's law [5]. In table 2. we compare the "information gain" scores obtained from both approaches. Information Gain is an established and empirically tested method for high-dimensional feature selection. It tells us the information contribution of a feature within a vector [24]. In our analysis, Information Gain score obtained from the "bag-of-words" approach for the unigram word "beautiful" is almost equal to the sum of the word "beautiful" and the n-gram "very beautiful". After repeating the same operation for other features as well, we have found that the "n-gram" approach does not provide efficient results for social media data. Hence, we have decided to proceed to the modeling phase with the "bag-of-words" approach.

Table 2. “Information Gain” scores for bag of words and N-Gram approaches

Ranked attributes of N-gram:			**Ranked attributes of bag of words**		
0.07414	11130	güzel	0.14844	403	güzel
0.07136	4191	bir	0.08109	1293	harika
0.07113	26802	çok güzel	0.07198	688	pahalı
0.05936	5013	bu	0.05936	201	bu
0.05771	18932	pahalı	0.05658	1466	olmuş
0.04908	25612	yok	0.0541	183	bir
0.04371	36	1	0.04908	940	yok
0.04358	27643	ürün	0.04614	869	var
0.04066	23712	ve	0.04389	981	çok
0.04041	23528	var	0.04371	4	1
0.03682	17380	neden	0.04358	1011	ürün
0.0364	11663	harika	0.0433	838	teşekkürler
0.03486	6543	da	0.04066	875	ve
0.03443	18367	olmuş	0.04033	783	sorun
0.03391	21444	sorun	0.03682	625	neden
0.03219	14812	ki	0.03352	1438	mükemmel
0.03076	246	2	0.03339	124	aynı
0.03076	13815	kadar	0.03219	533	ki
0.03076	1464	alışveriş	0.03076	81	alışveriş
0.02998	27190	çıktı	0.03076	502	kadar
0.02998	27102	çünkü	0.03076	14	2
0.02998	16020	lütfen	0.03072	701	pozitif
0.02998	4940	bozuk	0.03025	244	da
0.02998	2559	aynı	0.02998	574	lütfen
0.02885	12452	i	0.02998	986	çünkü
0.02772	6967	de	0.02998	197	bozuk
0.02659	18147	olduğunu	0.02998	987	çıktı
0.02633	10478	gibi	0.02885	268	değil

In the next step, we calculated the weight of each feature (word) according to the inverse document frequency (idf_t) transformation method. Idf_t is a measure of the relative importance of a term calculated as the concentration of a term in particular documents compared to its appearance in all the documents:

$$\text{idf}_t = \log \frac{N}{df_t} \tag{1}$$

Idf_t transformation shows us the relative frequency of a term compared to its appearance in all documents. Words with lower idf_t carry lower information values. For example, while function words like “the” have lower idf_t scores; opinion expression words like “beautiful” which are not distributed along all documents have higher scores.

In our study, we have tested the performance of the models including selection of features exceeding an idf$_t$ score threshold against a baseline including all features.

3.3 Modeling

We used Naïve Bayes and Support Vector Machines for classification of texts and compared the results with and without feature selection. Classification procedures were performed on positive and negative texts.

Table 3. Classification results for Brand-1 (355 texts)

	Accuracy rate	ROC area	Prec. (Pos)	Prec. (Neg)	Recall (Pos)	Recall (Neg)
Support Vector Mach	90.44%	0.905	0.924	0.883	0.893	0.917
Naive Bayes	92.53%	0.983	0.901	0.958	0.966	0.879
SVM (selected features)	85.97%	0.856	0.836	0.893	0.916	0.796
Naive Bayes (selected features)	91.44%	0.964	0.878	0.957	0.966	0.847

Table 4. Classification results for Brand-2 (425 texts)

	Accuracy rate	ROC area	Prec. (Pos)	Prec. (Neg)	Recall (Pos)	Recall (Neg)
Support Vector Mach	85.41%	0.789	0.838	0.923	0.972	0.606
Naïve Bayes	86.82%	0.912	0.856	0.909	0.969	0.657
SVM (selected features)	88.23%	0.823	0.858	0.968	0.99	0.657
Naïve Bayes (selected features)	86.82%	0.92	0.858	0.901	0.965	0.664

Models with feature selection were found to give more accurate results. We select the Naïve Bayes model with feature selection to build the sentiment classifiers. Naïve Bayes approach is more robust to the number of features, hence more efficient for detecting classes with the lower number of features (Table 3 and 4).

3.4 Results

The results from application of the Naïve Bayes model results on Brand-1 and Brand-2 are shown in Table 5 and 6.

Table 5. Application of the Naïve Bayes model results on Brand-1

Brand-1	Frequency	Percentage
Total	2834	
Negative	846	30%
Positive	1594	56%
Negative comments about the brands	90	10,6%
Positive comments about the brands	55	3,5%

Table 6. Application of the Naïve Bayes model results on Brand-2

Brand-1	Frequency	Percentage
Total	1906	
Negative	360	19%
Positive	1392	73%
Negative comments about the brands	52	14%
Positive comments about the brands	70	19%

As the tables suggest, Brand-2 products received more positive comments compared to Brand-1. Our results suggest that sentiment analysis can be an alternative or complementary method for attitude polls or surveys. Of course this approach carries some limitations in terms of sampling and instrument design. For example, the results of the analysis depend on the individuals who write comments on the Web pointing to a potential sampling bias. Moreover, measurement instrument (sentiment classifier) although highly reliable (giving the same results on the same sample) can pose some validity problems (measuring attitudes rather than other constructs such as idiosyncracies in language use). After rigorous cross-validating with other instruments like surveys and interviews, sentiment analysis has a potential for attitude measurement in the social media.

4 Conclusion

Application of sentiment analysis to the Turkish language and social media context is a challenging issue. Our results provide some evidence that with comprehensive preprocessing effort and machine learning algorithms, sentiment analyses can be effectively performed in Turkish Language. Future studies can improve the performance not only by analyzing the general attitudes towards brands, but also the attitudes towards specific brand attributes. Furthermore, adding a third neutral value to negative-positive poles can enrich the scale for precision purposes. The results of this study provide support to the argument that sentiment analysis can be used to complement survey based attitude research.

References

1. Krippendorf, K., Content Analysis; An Introduction to its Methodology, 3rd Edition, Thousand Oaks, CA: pp. 441, Sage (2012)
2. Shanahan, J., Qu, Y., & Wiebe, J. (Eds.). Computing attitude and affect in text: Theory and applications. Dordrecht, the Netherlands: Springer (2006)
3. Choi, Y., Breck, E., & Cardie, C. Joint extraction of entities and relations for opinion recognition. In Proceedings of Empirical Methods in Natural Language Processing (EMNLP). Sydney, Australia (2006)

4. Kim, S.-M., & Hovy, E. Extracting opinions expressed in online news media text with opinion holders and topics. In Proceedings of the Workshop on Sentiment and Subjectivity in Text at the Joint COLINGACL 2006 Conference pp. 1–8. Sydney, Australia (2006)
5. Pang, Bo, Lee, Lillian, and Vaithyanathan, Shivakumar.. Thumbs up? Sentiment classification using Machine Learning techniques. In Proceedings of Conference on Empirical Methods in NLP, pp. 79-86 (2002)
6. Tong, Richard M. An operational system for tracking opinions in on-line discussions. In Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification, 1-6. New York, NY: ACM. (2001)
7. Dave, Kushal, Lawrence, Steve, and Pennock, David M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the Twelfth International World Wide Web Conference (WWW 2003). Budapest, Hungary (2003)
8. Mullen, Tony, and Collier, Nigel. Sentiment analysis using support vector machines with diverse information sources. In Proceedings of Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain (2004)
9. Whitelaw, Casey, Garg, Navendu, and Argamon, Shlomo.. Using Appraisal groups for sentiment analysis. In Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM 2005), pp. 625-631. Bremen, Germany (2005)
10. Turney, Peter. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of 40th Meeting of the Association for Computational Linguistics, pp. 417-424 (2002)
11. Taboada, Maite, and Grieve, Jack. Analyzing appraisal automatically. In Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS-04-07), eds. Yan Qu, James G. Shanahan and Janyce Wiebe, 158-161. Stanford University, CA: AAAI Press. (2004)
12. Bloom, Kenneth, Garg, Navendu, and Argamon, Shlomo. Extracting Appraisal Expressions. Paper presented at NAACL HLT 2007, Rochester, NY. (2007)
13. Breck, E., Choi, Y., & Cardie, C. Identifying expressions of opinion in context. In M. M. Veloso (Ed.), Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). Hyderabad, India. (2007)
14. Wilson, T., Wiebe, J., & Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing HLT-EMNLP-2005. Vancouver, Canada (2005)
15. Witten, Ian H., and Frank, Eibe. Data Mining: Practical Machine Learning Tools and Techniques. San Francisco: Morgan Kaufmann. (2005)
16. Mihalcea, R., Banea, C., & Wiebe, J. Learning multilingual subjective language via cross-lingual projections. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07) pp. 968–975, Prague, Czech Republic (2007)
17. Mathieu, Y. A computational semantic lexicon of French verbs of emotion. In J. Shanahan, Y. Qu, & J. Wiebe (Eds.), Computing attitude and affect in text: Theory and applications pp. 109–123, Dordrecht, the Netherlands: Springer. (2006)
18. Cho, Young Hwan, and Lee, Kong Jo. Automatic affect recognition using natural language processing techniques and manually built affect lexicon. IEICE Transactions on Information and Systems 89: 2964-2971. (2006)
19. Abbasi, Ahmed, Chen, Hsin-Hsi, and Salem, Arab. Sentiment Analysis in Multiple Language: Feature Selection for Opinion Classification in Web Forums. ACM Transactions on Information Systems 26. (2008)

20. Kanamaru, T., Murata, M., & Isahara, H. Japanese opinion extraction system for Japanese newspapers using machine-learning method. In Proceedings of NTCIR-6 Workshop Meeting. Tokyo, Japan. (2007)
21. Xu, R., Wong, K.-F., & Xia, Y. Opinmine— opinion analysis system by for NTCIR-6 pilot task. In Proceedings of NTCIR-6 Workshop Meeting. Tokyo, Japan. (2007)
22. Boynukalın, Z. Emotion analysis of Turkish texts by using machine learning methods. Middle East Technical University. (2012)
23. Eryigit, G., Cetin, F. S., Yanık, M., Bilgi, T. G., Temel, T., & Çiçekli, I. Turksent: A sentiment annotation tool for social media. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse pp. 131-134, (2013)
24. Forman, G. An extensive empirical study of feature selection metrics for text classification JMLR 3 1289-1306 (2003)

Sentiment Lexicon-Based Features for Sentiment Analysis in Short Text

Hussam Hamdan,[1,2,3] Patrice Bellot,[1,2] Frederic Bechet[3]

[1] Aix Marseille Université, CNRS, ENSAM,
[2] Université de Toulon, LSIS UMR 7296,13397, Marseille, France
hussam.hamdan,patrice.bellot@lsis.org
[3] Aix Marseille Université, CNRS, LIF UMR 7279, Marseille, France
frederic.bechet@lif.univ-mrs.fr

Abstract. Sentiment lexicon-based features have proved their performance in recent work concerning sentiment analysis in Twitter. Automatic constructed lexicon features seem to be enough influential to attract the attention. In this paper, we propose a new metric to estimate the word polarity score, called natural entropy (ne), in order to construct a new sentiment lexicon based on Sentiment140 corpus. We derive six features from the new lexicon and show that (ne) metric outperforms the PMI metric which has been used for the same purpose. For evaluation, we build a state-of-the-art system for sentiment analysis in short text using a supervised classifier trained on several groups of features including n-gram, sentiment lexicons, negation, Z score and semantic features. This system has been one of the best systems in both tasks of SemEval-2015: Sentiment Analysis in Twitter and Aspect-Based Sentiment Analysis. We investigate the impact of the lexicon-based features extracted from existing manual and automatic constructed lexicons on the system performance and also the impact of the proposed metric (ne).

1 Introduction

The interactive Web has changed the relation between the users and the web. Users have become an important source of content. This content includes users' opinions about events, products, or people. The availability of such information has attracted growing attention from those who want to understand the opinions and preferences of individuals which may be useful in various domains.

Sentiment Analysis (SA) has become more and more interesting since the year 2000 [1], many techniques in Natural Language Processing have been used to understand the expressed sentiment on an entity. The basic task in sentiment analysis is the polarity classification which determines the polarity of a given text, i.e. whether the expressed opinion is positive, negative or neutral. This analysis can be done at different levels of granularity: Document Level, Sentence Level or Aspect Level.

Early work in that area includes [2] and [3] applied different methods for detecting the polarity; they proposed unsupervised, semi-supervised and supervised

pp. 217–226; rec. 2015-02-01; acc. 2015-02-27

methods. The document representation is a critical component in SA, several publications have focused on the term weighting, others on the feature extraction and selection.

In this paper, we focus on the impact of automatic constructed lexicons on the supervised sentiment classification in short text. Therefore, we build a new automatic sentiment lexicon using new metric called natural entropy (ne) which has recently been proposed for supervised term weighting [4], this lexicon is built from sentiment140 corpus [5] which contains 1.6 millions automatically collected tweets classified as positive or negative depending on the involved emoticons. After getting this new lexicon, six features have been extracted and have replaced the six features extracted from sentiment140 lexicon, which is built from the same sentiment140 corpus but using PMI (pair-wise mutual information) metric instead. The new (ne) lexicon features outperform those extracted from the original PMI sentiment140 lexicon. For evaluation purpose, we use two known data set, the first extracted from twitter, the second from restaurant reviews. We build two state-of-the-art systems using a Logistic Regression classifier with different types of features including word n-gram, twitter dictionary, Z score, sentiment lexicon and semantic features, we also adapts a weighting schema for tuning the parameters of our classifiers. These two systems are among the best systems participating in SemEval 2015 task of Sentiment Analysis in Twitter and Aspect-based Sentiment Analysis. We investigate the effect of sentiment lexicon features in the two classifiers, as these features play an import role, we have been enough motivated to propose the using of (ne) metric.

The rest of this paper is organized as follows. Section 2 outlines existing work in sentence level sentiment analysis. Section 3 describes the data and resources that have been used. The features we used for representing the document are presented in Section 4. Our experiments are described in section 5, and future work is presented in Section 6.

2 Related Work

Two main approaches for sentiment analysis can be identified. The lexicon based approach which depends on sentiment lexicons containing positive, negative and neutral words or expressions; the polarity is computed according to the number of common opinionated words between the lexicons and the text. Many lexicons have been created manually such as MPQA Subjectivity Lexicon [6] or automatically such as SentiWordNet [7].

The second one is the Machine Learning approach which adapts different classifiers and features. Naive Bayes, Maximum Entropy MaxEnt and Support Vector Machines (SVM) were adapted in [5], the authors reported that SVM outperforms other classifiers, they tried a unigram and a bigram model in conjunction with part-of-speech (POS) features; they noted that the unigram model outperforms all other models when using SVM and that POS features decrease the results. The Authors in [8] found that n-gram with lexicon features and microbloging features are useful but POS features are not. In contrast, in [9] the

authors reported that POS and bigrams both help. In [10] the authors proposed the use of specific features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS tags. Authors in [11] used the concepts extracted from DBPedia and the adjectives from WordNet, they reported that the DBpedia concepts are useful with Naïve-Bayes classifier but less useful with SVM. Many features were used with SVM including the lexicon-based features in [12] which seem to get the most gain in performance. Other work has also proved the importance of lexicon-based features with logistic regression classifier [13, 14].
Some sentiment lexicons are manually constructed. A label (positive, negative) or a polarity score is assigned by human annotators. While automatically constructed ones assign a score indicating the association with the positive or negative sentiment, this score is computing with the aid of an annotated corpus or a linguistic resource. These scores make us capable of ranking the terms according to their association to the sentiment. Authors in [2] estimated the sentiment orientation (SO) of the extracted phrases using the Pointwise Mutual Information (PMI). The sentiment orientation of a phrase is computed based on its association with the positive reference word "excellent" and the negative reference word "poor". Authors in [15] used SO to compute the sentiment orientation of a given word. In [16] Authors collected a set of 775,000 tweets to generate a large word-sentiment association lexicon; a tweet was considered positive if it has one of 32 positive hashtagged seed words, and negative if it has one of 36 negative hash-tagged seed words; the association score for a term was calculated using SO. Authors in [12] used similar method on the sentiment140 corpus [5], a collection of 1.6 million tweets that contains positive and negative emoticons; the tweets are labeled positive or negative according to the emoticons.

3 Data and Resources

3.1 Training and Testing Data

We have used two data sets, the first one is extracted from Twitter which has been provided in SemEval 2013 for subtask B of sentiment analysis in Twitter [17]. The participants have been provided with training tweets annotated positive, negative or neutral. We downloaded these tweets using the given script. We obtained 9646 tweets, the whole training data set is used for training, the provided development set containing 1654 tweets is used for tuning the machine learner. The test data set provided in SemEval-2015 containing about 2390 tweets [18] is used for evaluating our system.
The second data set is extracted from restaurant reviews, provided by SemEval 2015 ABSA organizers [19] where each review is composed of several sentences and each sentence may contain several Opinion Target Expression OTE which we want to detect their polarities. Table 1 shows the distribution of each label in each data set.

Table 1. Sentiment labels distribution in the training, testing and development data sets in Twitter and Restaurant.

Data	All	Positive	Negative	Neutral
Twitter				
train	9684	3640	1458	4586
dev	1654	739	340	575
test	2390	1038	365	987
Restaurant				
train	1655	1198	403	53
test	845	454	346	45

3.2 Sentiment Lexicons

In this section, we describe the manual and automatic constructed sentiment lexicons which have been used for realizing our system, and also explain the new lexicon we have constructed using the natural entropy metric.

Manual Constructed Sentiment Lexicons:

1- MPQA Subjectivity Lexicon: MPQA (Multi-Perspective Question Answering) Subjectivity Lexicon is maintained by Theresa Wilson, Janyce Wiebe, and Paul Hoffmann [6], a lexicon of over 8,000 subjectivity single-word clues, each clue is classified as positive or negative.

2- Bing Liu Lexicon: A list of positive and negative opinion words or sentiment words for English (around 6800 words). This list was compiled over many years starting from this paper [20].

Automatic Constructed Sentiment Lexicons:

1- NRC Hashtag Sentiment Lexicon: NRC Hashtag Sentiment Lexicon [16] contains tweet terms with scores, positive score indicates association with positive sentiment, whereas negative score indicates association with negative sentiment. It has entries for 54,129 unigrams and 316,531 bigrams; the scores are computed using PMI over a corpus of tweets.

2- Sentiment140 Lexicon: Sentiment140 Lexicon [12] contains tweet terms with scores, Sentiment140 has entries for 62,468 unigrams, 677,698 bigrams. the scores are computed using PMI over sentiment140 tweet corpus .

3- SentiWordNet: SentiWordNet [7] is the result of automatically annotating all WORDNET synsets according to their degrees of positivity, negativity, and neutrality.

4- Our Lexicon: PMI metric has been widely used to compute the semantic orientation (SO) of words in order to construct the automatic lexicons. Sentiment140 lexicon is constructed using SO but Sentiment140 corpus is a balanced corpus, it contains the same number of positive and negative tweets. Therefore, SO can be rewritten as following:

$$SO(w) = PMI(w,+) - PMI(w,-) = log(\frac{p(w,+)}{p(w).p(+)}) - log(\frac{p(w,-)}{p(w).p(-)})$$

As $p(+) = p(-)$=0.5 in the balanced corpus:

$$So(w) = 1 + log(p(+|w)) - 1 - log(p(-|w)) = log(a/c)$$

where + stands for the positive class, - stands for negative class, a is the number of documents containing the word w in the positive class, c is the number of documents containing the word w in the negative class. Thus, the SO is positive if a>c else it is negative. We should note that the probability of the classes does not affect the final SO score, therefore we propose another metric which depends on the distribution of the word over the classes which seems more relevant in the balanced corpus. We constructed a lexicon from sentiment140 corpus [5] a collection of 1.6 million tweets that contain positive and negative emoticons, we calculated Natural Entropy score for each term in this manner:

$$ne(w) = 1 - (-(p(+|w).log(p(+|w)) - p(-|w).log(p(-|w)))) \qquad (1)$$

where p(+| w): The probability of the positive class given the word w, p(−| w): The probability of the negative class given the word w. The more uneven the distribution of documents where a term occurs, the larger the Natural Entropy (ne) of this term is. Thus, the entropy of the term can express the uncertainty of the classes given the term. One minus this degree of uncertainty boosts the terms that unevenly distributed between the two classes [4]. ne score is always between 0 and 1, and it assigns a high score for the words unevenly distributed over the classes, but it cannot discriminate the positive words from the negative ones. therefore, we have used the a and c for discriminating the positive words from the negative ones; if a>c then the word is considered positive else it is considered negative.

3.3 Twitter dictionary

We constructed a dictionary for the abbreviations and the slang words used in Twitter in order to overcome the ambiguity of these terms. This dictionary maps certain twitter expressions and emotion icons to their meaning or their corresponding sentiment (e.g. gr8 replaced by great, :) replaced by very-happy).

4 Feature Extraction

Text representation for sentiment analysis can be enriched by many types of features. Adding syntactic and semantic features may result in important improvement on the system performance. In this section, we present several groups of features which have proved their performance in many experiments in SemEval 2015.

Word n-grams

unigram and bigram are extracted for each word in text without any stemming or stop-word removing, all terms with occurrence less than 3 are removed from the feature space.

Negation features

The rule-based algorithm presented in Christopher Potts' Sentiment Symposium Tutorial is implemented. This algorithm appends a negation suffix to all words that appear within a negation scope which is determined by a negation key and a punctuation. All these words have been added to the feature space.

Twitter Dictionary

All terms presented in the text and in the twitter dictionary (presented in 3.3) are mapped to their corresponding terms in the dictionary and added to the feature space.

Sentiment Lexicons

The system extracts four features from the manual constructed lexicons and six features from the automatic ones. For each sentence the number of positive words, the number of negative ones, the number of positive words divided by the number of negative ones and the polarity of the last word are extracted from manual constructed lexicons. In addition to the sum of the positive scores and the sum of the negative scores from the automatic constructed lexicons.

Z Score

Z score can distinguish the importance of each term in each class, their performances have been proved in [21]. We assume as in the mentioned work that the term frequencies are following a multi-nomial distribution. Thus, Z score can be seen as a standardization of the term frequency using multi-nomial distribution. We compute the Z score for each term ti in a class C_j (t_{ij}) by calculating its term relative frequency tfr_{ij} in a particular class C_j, as well as the mean $(mean_i)$ which is the term probability over the whole corpus multiplied by the number of terms in the class C_j, and standard deviation (sd_i) of term ti according to the underlying corpus. Like in [11] we tested different thresholds for choosing the words which have higher Z score.

$$Zscore(ti) = \frac{tfr_{ij} - mean_i}{sd_i} \tag{2}$$

Thus, we added the number of words having Z score higher than the threshold in each class positive,negative and neutral, the two classes which have the maximum number and minimum number of words having Z score higher than the threshold. These 5 features have been added to the feature space.

Semantic Features

The semantic representation of a text may bring some important hidden information, which may result in a better text representation and a better classification system.

-Brown dictionary features Each word in the text is mapped to its cluster in Brown, 1000 features are added to feature space where each feature represents

the number of words in the text mapped to each cluster. The 1000 clusters is provided in Twitter Word Clusters of CMU ARK group. 1000 clusters were constructed from approximately 56 million tweets.
-Topic Features Latent dirichlet association or topic modeling is used to extract 10 features. Lda-c is configured with 10 topics and the training data is used for training the model, then for each sentence in the test set, the trained model estimates the number of words assigned to each topic.
-Semantic Role Labeling Features Authors in [22] encode semantic role labeling features in SVM classifier. Our system also extract two types of features, the names: the whole term which represents an argument of the predicate and the tags: the type of each argument in the text (A0 represents the subject of predicate, A1 the object, AM-TMP the time, AM-ADV the situation, AM-loc the location). These encodings are defined by the tool which we used (Senna). We think that the predicate arguments can constitute a multi-word expression which may be helpful in Sentiment Classification.

5 Experiments

5.1 Experiment Setup

We used L1-regularized Logistic regression classifier implemented in LibLinear [23], this classifier has given good results in recent work [24] [14]. We learned two classifiers one from twitter data set using all features of Section 4 with the three polarities (positive, negative, and neutral) as labels and the second from restaurant review data set using only the following features (word n-gram, negation, lexicon-based, Z score, Brown cluster). A weighting schema is adapted for each class, we use the weighting option $-w_i$ which enables a use of different cost parameter C for different classes. Since the training data is unbalanced, this weighting schema adjusts the probability of each label. Thus, we tuned the classifier in adjusting the cost parameter C of Logistic Regression, weight wpos of positive class and weight Wneg of negative class. We used the twitter development set and 10% from the training data of restaurants for tuning the three parameters, all combinations of C in range 0.1 to to 4 by step 0.1, w_{pos} in range 1 to 8 by step 0.1, w_{neg} in range 1 to 8 by step 0.1 are tested. The combination C=0.2, w_{pos}=5.2, w_{neg}=4.2 has given the best F1 score on the development set of Twitter data set and the combination C=0.3, w_{pos}=1.2, w_{neg}=1.9 has been chosen for Restaurant set.

5.2 Results

The evaluation score used for twitter data set is the averaged F1-score of the positive and negative classes as proposed by the task organizers [17] but the averaged F1-score of all classes for restaurant review. Table 2 shows the results of our experiments after removing one lexicon features at a time for the two test sets besides to the experiment which evaluates the effect of using our sentiment lexicon which based on (ne) metric instead of sentiment140 lexicon which based on PMI metric. Note that using the lexicon features provides a gain of 3.31%, 4.50% for the twitter and restaurant test sets respectively. The manual lexicon features provide a gain of 0.98%, 0.47%, the automatic lexicon ones provide 1.13%, 1.06% which seem to be more influential than the manual ones. The results after removing each lexicon features shown in tabel 2. Note that some lexicon features decrease the performance in restaurant review such as Big-Liu, sentiment 140 and SentiwordNet but all lexicon features are influential in twitter set, the MPQA and SentiwordNet are the less influential. The last line of table 2 shows the results after removing the sentiment140 lexicon features but adding our lexicon features instead. Our features improve the performance by 0.62%, 0.48% on the two data set. This extrinsic evaluation is an indicator that using (ne) metric can be more efficient than using PMI for building a sentiment lexicon.

Table 2. The F-scores obtained on the Twitter and Restaurant test sets, Allfeatures run exploits all proposed features (see 5.1), all-lexicons run removes the lexicons features from the first run (the whole feature space), all-automatic run removes the automatic lexicon features, all-manual, all-MPQA, all-BingLiu, NRC-Hashtag, all-Sentiment140, all-SentiWordNet remove the manual lexicons, MPQA lexicon, Big LIU, NRC Sentiment140, and SentiWordNet respectively from the whole feature space, the last run removes the features extracted from Sentiment140 but adds those extracted from our new lexicon instead.

	Tweet Test	**Rest Test**
Allfeatures	64.27	75.50
all-lexicons	60.96	71.00
all-automatic	63.14	74.44
all-manual	63.29	75.03
all-MPQA	64.11	75.27
all-Bing Liu	63.69	76.33
all-NRC Hashtag	63.90	74.67
all-Sentiment140	63.91	75.62
all-SentiWordNet	64.08	75.60
all-Sentiment140+Our lexicon	64.89	75.98

6 Conclusion and future work

We built two state-of-the-art classifiers for sentiment analysis in short text. One for Twitter data and other for restaurant reviews. We study the impact of lexicon-based features on the performance. We also constructed our own sentiment lexicon using new metric called natural entropy (ne) which boosts the terms that unevenly distributed among the classes. This new lexicon features seem to improve the results more than the features extracted from the same

lexicon but using PMI metric.
As the sentiment lexicon-based features have proved their performance, future work will focus on the automatic lexicon construction on testing several metrics like Z score and KL-Divergence which we think promising in measuring the association between terms and sentiment labels.

7 Acknowledgment

This work was supported by Région PACA, Agoraweb projet.

References

1. Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.
2. Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424. Association for Computational Linguistics, 2002.
3. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86. Association for Computational Linguistics, 2002.
4. Haibing Wu and Xiaodong Gu. Reducing over-weighting in supervised term weighting for sentiment analysis. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1322–1330, 2014.
5. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. pages 1–6, 2009.
6. Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, pages 34–35. Association for Computational Linguistics, 2005.
7. Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of LREC*, 2010.
8. Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the OMG! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press, 2011.
9. Er Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *In Proceedings of the Seventh Conference on International Language Resources and Evaluation*, 2010.
10. Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44. Association for Computational Linguistics, 2010.

11. Hussam Hamdan, FrÃ©dÃ©ric Bechet, and Patrice Bellot. Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In *International Workshop on Semantic Evaluation SemEval-2013 (NAACL Workshop)*, 2013-04-29.
12. Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. NRCCanada: Building the state-of-the-art in sentiment analysis of tweets. In *In Proceedings of the International Workshop on Semantic Evaluation, SemEval '13*, 2013.
13. Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. TeamX: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632. Association for Computational Linguistics and Dublin City University, 2014-08.
14. Hussam Hamdan, Patrice Bellot, and Frederic Bechet. lsislif: Feature extraction and label weighting for sentiment analysis in twitter. In *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.
15. Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. 21(4):315–346, 2003-10.
16. Saif Mohammad. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255. Association for Computational Linguistics, 2012-06-07.
17. Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. SemEval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320. Association for Computational Linguistics, 2013.
18. Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M. Mohammad, Alan Ritter, and Veselin Stoyanov. SemEval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '2015. Association for Computational Linguistics, 2015-06.
19. Maria Pontiki, Dimitrios Galanis, Haris Papageogiou, Suresh Manandhar, and Ion Androutsopoulos. SemEval-2015 task 12: Aspect based sentiment analysis. In *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.
20. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177. ACM, 2004.
21. Hussam Hamdan, Patrice Bellot, and Frederic Bechet. The impact of z_score on twitter sentiment analysis. In *In Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval 2014)*, page 636, 2014.
22. Josef Ruppenhofer and Ines Rehbein. Semantic frames as an anchor representation for sentiment analysis. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 104–109. Association for Computational Linguistics, 2012.
23. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. 9:1871–1874, 2008.
24. Hussam Hamdan, Patrice Bellot, and Frederic Bechet. Lsislif: CRF and logistic regression for opinion target extraction and sentiment polarity analysis. In *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

Automatic Aspect Identification: The Case of Informative Microaspects in News Texts

Alessandro Y. Bokan[1 2] and Thiago A. S. Pardo[1 2]

[1] Institute of Mathematical and Computer Sciences, University of São Paulo, SP, Brazil
[2] Interinstitutional Center for Computational Linguistics, São Carlos, SP, Brazil
{abokan,taspardo}@icmc.edu.br

Abstract. Informative aspects represent the basic units of information in texts. For example, in news texts they could represent the following information: what happened, when it happened and where it happened. With the identification of these aspects, it is possible to automate some NLP tasks such as Summarization, Question Answering and Information Extraction. *Microaspects* --a type of informative aspects-- represent local segments of the sentence. In this paper, we automatically identify *microaspects* using Semantic Role Labeling, Named-Entity Recognition, Handcrafted Rules and Machine Learning techniques. We evaluate our proposal using the CSTNews journalistic corpus, which has manually annotated aspects. The results are satisfactory, and prove that *microaspects* can be automatically identified in news texts with acceptable performance.

Keywords: Automatic Summarization, Semantic Role Labeling, Named-Entity Recognition, Machine Learning

1 Introduction

Informative aspects represent semantic-discursive basic units of information present in sentences. Aspects can represent local components of a sentence, like specific location or a certain date. They can also appear from the sentence context. For example, in natural disasters news, the following aspects could be recognized: when it happened?, where it happened?, how it happened?, and what happened?.

Aspects date back to the work of Swales [14] with the model CARS (Create a Research Space), where schematic structures are used to create/organize scientific texts. Recently, the TAC (Text Analysis Conference) – the main conference and scientific competition about Automatic Summarization – proposed the use of informative aspects to assist the Summarization Guided task (2010/2011). For example, TAC analyzed and determined that summaries of the category "Attacks" might include specific aspects as: what happened (WHAT), casualties (WHO_AFFECTED), perpetrators (PERPETRATORS), location (WHERE) and date (WHEN).

Fig. 1 presents an aspect annotation of a summary of the category "Attacks". The first sentence reports that several attacks occurred (WHAT) in São Paulo (WHERE), on Monday (WHEN). The second sentence identifies the entities (person or organization) affected by the attacks (WHO_AFFECTED). Finally, the last sentence identifies the criminal entities (PERPETRATORS).

[A new series of criminal attacks happened at dawn of Monday, March 7, in São Paulo and near cities.] **WHAT/WHEN/WHERE**
[The bandits attacked banks, police stations and public buildings with bombs and gunshots.] **WHO_AFFECTED**
[Those actions are attributed to the criminal gang "Primeiro Comando da Capital" (PCC), which has led twice others attacks.] **PERPETRATORS**

Fig. 1. Example of aspects annotation

The main goal to identifying aspects is to **automate** some NLP tasks such as Summarization, Question Answering and Information Extraction. According to Genest et al. [7], "*aspect identification can be useful to determine relevant information from source texts and to identify structural constraints to construct summaries"*. Owczarzak and Dang [10] proposed the use of informative aspects as a deep approach to produce coherent and cohesive multi-document summaries for a specific genre (*e.g.,* journalistic, narrative, opinion, etc.) and category (*e.g.,* sports, politics, etc.). Since TAC, aspects have been used in several studies in the literature to assist the summarization task [13,8].

Aspects can indicate standard structures (*templates*) to model criteria of content selection and organization to automatically generate coherent summaries. For that reason, Rassi et al. [12] manually annotated aspects on multi-document summaries in the Brazilian Portuguese journalistic corpus called CSTNews [6].

CSTNews was built mainly to assist the Multi-document Automatic Summarization task. The identification of textual segments in different sentential structural levels to determine informative aspects resulted in the classification of aspects in *microaspects* and *macroaspects*. *Microaspects* represent local segments that make up a sentence. *Macroaspects* emerge from the combination of linguistic patterns contained in the local segments inside a sentence, or from the relationship between two or more sentences.

In this paper we use two approaches to automatically identify *microaspects*. The first approach is based on semantic roles, named-entities and handcrafted rules. The second approach is based on machine learning techniques. We evaluate our proposal using a set of aspect-annotated summaries in the CSTNews corpus. In this work, aspects are specifically defined for **journalistic genre**, based on the TAC's Summarization task.

The remaining of the paper is organized as follows: in Section 2, we introduce some related work; in Section 3, we describe the two approaches used to identify *microaspects*; the experiments and results are presented in Section 4; finally, in Section 5, we conclude this paper.

2 Related Work

2.1 Text Analysis Conference

The “Text Analysis Conference” (TAC) is the main conference and scientific competition about Automatic Summarization (AS). In 2010[1], aspects have been proposed to assist the Guided Summarization task to explore a deeper linguistic analysis of the source documents. The goal was to generate a 100-word summary from a set of 10 "newswire" articles for a given topic. Each topic falls into a predefined category. All the participants in the task were given a list of aspects for each category. Finally, the generated summary should include all aspects defined for its category. Table 1 shows some categories and their defined aspects. The remainder categories are: “Health and Safe”, “Endangered Resources” and “Trials and Investigations”.

Table 1. Aspects defined for each category

Category	Aspects
Accidents and Natural Disasters	What happened (WHAT); date (WHEN); location (WHERE); rescue efforts (COUNTERMEASURES); damages caused by the accident/disaster (DAMAGES); reasons for accident/disaster (WHY); casualties (WHO_AFFECTED).
Attacks	What happened (WHAT); date (WHEN); location (WHERE); reasons (WHY); casualties (WHO_AFFECTED); entity responsible for the attack (PERPETRATORS); damages caused by the attack (DAMAGES); rescue efforts (COUNTERMEASURES).

Some studies were done using TAC 2010 principles. Steinberger et al. [13] proposed a deep semantic analysis to model informative aspects for multilingual AS. Makino et al. [9] and Li et al. [8] automatically identified informative aspects in Wikipedia and generated summaries based on those aspects. Barrera et al. [3] created a question-answering system, called *SemQuest*, based on aspect identification for different categories. Even before TAC, some works presented similar approaches, for example, White et al. [15] proposed aspects-based templates for summaries of “Natural Disasters” texts.

2.2 CSTNews Corpus

The CSTNews corpus [6] is a resource that contains 50 Brazilian Portuguese journalistic text collections. Each collection has 2-3 documents on the same subject but from different sources. The collections were classified into 6 textual categories: Daily News (14), Sports (10), World (14), Politics (10), Money (1) and Science (1). In addition to the raw texts, CSTNews comprises 140 manually generated single-document abstractive summaries, 50 manually generated multi-document abstractive summaries and 50 manually generated multi-document extractive summaries.

[1] http://www.nist.gov/tac/2010/Summarization/

Rassi et al. [12] annotated aspects over the 50 manually generated multi-document abstractive summaries from CSTNews corpus. CSTNews categories are different from TAC-2010 categories. However, there are similarities between categories, *e.g.,* “Daily news” and “World” could contain “Accidents and natural disasters” topics. As mentioned before, aspects were divided in *microaspects* and *macroaspects*. *Microaspects* represent local segments that make up a sentence. *Macroaspects* emerge from the combination of linguistic patterns contained in the local segments inside a sentence, or from the relationship between two or more sentences. In total, [12] identified 8 *microaspects* (see Table 2). It is important to say that aspects were annotated at the **end of the sentence** (sentential level).

Table 2. CSTNews *microaspects* definition

Microaspect	**Definition**
WHO_AGENT	Entity (person/organization) responsible for the fact/event
WHO_AFFECTED	Entity (person/organization) affected by the fact/event
WHEN	Date/time of occurrence of the fact/event
WHERE	Physical/geographical location of the fact/event
WHY	Why the fact/event happens (reasons)
HOW	How the fact/event occurs
SCORE	Result of the sport event
SITUATION	Situation when the fact/event occurs

Fig. 2 shows an example of an annotated sentence with aspects of the “World” category. Concerning to *macroaspects*, it is reported a natural disaster event (WHAT) and the declaration emitted by the pro-Pyongyang Japanese newspaper (DECLARATION). On the other hand, concerning to *microaspects*, it is reported that the disaster happened in July (WHEN), in North Korea (WHERE), because of the floods (WHY) and leaving many dead and injured people (WHO_AFFECTED).

> [A study from the japanese newspaper that supports pro-Pyongyang says that, in decorrence of the floods that hit North Korea in july, at least 549 people passed away, 3.043 people were wounded and 295 people are missing.]**WHAT,WHEN, WHO_AFFECTED,WHY,WHERE,DECLARATION**

Fig. 2. Annotated sentence of a summary in the CSTNews corpus

2.3 PALAVRAS Parser

PALAVRAS is a rule-based syntactic parser for Portuguese developed by [4]. In addition, it produces a list of semantic tags[2]. It has two output formats: a simple format (“flat”), and a traditional syntactic tree format (“tree”).

According to its author, PALAVRAS achieved a correctness rate of over 99% for morphology and part-of-speech. For syntax the figures are 97-98%. In this work, we

[2] http://beta.visl.sdu.dk/visl/pt/info/portsymbol.html#semtags_nouns

use PALAVRAS as input for the semantic role classifier (*tree* format) and to create an aspect classifier using Machine Learning techniques (*flat* format).

2.4 Semantic Role Labeling

Semantic roles represent the semantic relationships between verbs and their arguments. The task of identifying which phrases act as arguments of a particular verb is called Semantic Role Labeling (SRL). For Brazilian Portuguese, [1] proposed a supervised classification system that consists of 3 phases: (1) target verb identification, (2) argument identification and (3) argument classification.

Fig. 3 shows an example of the SRL process. Firstly, the target verb "won" is identified. Secondly, arguments (*A*) are identified: "Brazilian team", "Finnish team" and "in Tampere". Lastly, arguments are annotated with semantic roles: A0 (agent), A1 (patient) and AM-LOC (location), respectively. The "A" terminology refers to the "argument" followed by a prototypical number (1-5), and the "AM" terminology refers to the "modifier argument", followed by a type of modifier, such as time, location, cause, etc. Semantic roles were defined by [11]. It achieved a F1 measure of 94.5% in the identification phase and 81.70% in the classification phase, being the best system for Brazilian Portuguese.

The Brazilian team [won] over the Finnish team in Tampere. (1)
[The Brazilian team]$_{A}$ [won] [over the Finnish team]$_{A}$ [in Tampere]$_{A}$. (2)
[The Brazilian team]$_{A0}$ [won] [over the Finnish team]$_{A1}$ [in Tampere]$_{\text{AM-LOC}}$. (3)

Fig. 3. Semantic Role Labeling annotation example

Arguments related to the verb can answer some questions like who, where, when, why and how. In the previous example, the answers to the questions who won?, who lost? and where won?, are "Brazilian team", "Finland team" and "Tampere", respectively. Therefore, semantic roles can define informative aspects such as WHO_AGENT (who won?), WHO_AFFECTED (who lost?) and WHERE ("where won?). In this paper we propose the use of the SRL system for Brazilian Portuguese developed by [1] to identify some *microaspects*. Table 3 presents the proposed equivalences among some *microaspects* and the corresponding semantic roles.

Table 3. Equivalences among *microaspects* and semantic roles.

Microaspect	Semantic role	Name	Definition
WHO_AGENT	A0	Agent	Subject who did the action
WHO_AFFECTED	A1	Patient	Subject affected by the action
WHERE	AM-LOC	Location	Where the action happens
WHEN	AM-TMP	Temporal	When the action happens
HOW	AM-MNR	Manner	How the action was performed
WHY	AM-CAU	Cause	Reasons for the action

2.5 Named-entity Recognition (NER)

Named-entities (NE) are concrete or abstract entities referenced in the text by a proper noun. Named-Entity Recognition (NER) is a subtask of Information Extraction that aims to identify and classify text entities of predefined categories such as LOCATION, TIME, and EVENT, among other categories of interest.

HAREM[3] is an evaluation event of NER systems for Portuguese document collections. Many works were presented in the two editions of the HAREM. One of the best open-source systems, presented in the second HAREM, was REMBRANDT [5]. REMBRANDT intensely explores Wikipedia as a knowledge source and executes a set of grammatical rules to take advantage of the internal and external indications of the NEs to extract its significance. Furthermore, REMBRANDT has a proper interface to interact with Wikipedia, SASKIA, that provides simple category navigation. That system has 56.74% F1 measure for NER.

NE categories can define some informative aspects: WHERE is equivalent to LOCATION, WHEN to TIME, and SITUATION to EVENT. Thus, we propose the use of REMBRANDT system to automatically identify some *microaspects*. Table 4 shows the proposed equivalences among *microaspects* and NE categories.

Table 4. Equivalences among *microaspects* and NE categories

Microaspect	NE Category
WHERE	LOCATION
WHEN	TIME
SITUATION	EVENT

3 Methodology

The automatic identification process was divided in three phases: (1) to compile all the 322 sentences of the 48 CSTNews annotated summaries of categories Daily, Sports, World and Politics; (2) to automatically annotate sentences with *microaspects* using the 3 proposed systems, called SRL, SRL+Rules, REMBRANDT; and Machine Learning techniques; (3) finally, to get a set of annotated sentences.

3.1 SRL System

The SRL system automatically annotates *microaspects* equivalent to semantic roles (*cf.* Table 3). It covers WHO_AGENT, WHO_AFFECTED, WHEN, WHERE, WHY, and HOW. Given a set of sentences, we use PALAVRAS parser *tree* format to generate syntactic trees for each sentence. Trees are represented in *TigerXML* format. Auxiliary verbs were not considered. Following, each instance was annotated by the Alva-Manchego [1] classifier. Semantic roles are then mapped to the corresponding *microaspects* (see Fig. 4). Finally, all annotated aspects are positioned at the **end** of the sentence.

[3] http://www.linguateca.pt/harem/

<aspect SRL="WHO_AGENT"> The Brazilian team </aspect> **won** <aspect SRL="WHO_AFFECTED"> over the Finnish team </aspect> <aspect SRL= "WHERE"> in Tampere </aspect>.

Fig. 4. *Microaspect* sentence annotation using SRL system

3.2 SRL+Rules System

The SRL+Rules system uses handcrafted rules based on patterns founded in **false negatives** and **false positives** of the SRL system, in order to improve its performance. Rules were created for aspects WHO_AGENT, WHO_AFFECTED, WHEN, WHERE and WHY. Also, SCORE rules were defined, despite not existing equivalence with any semantic role. It is crucial to specify that all rules were **specifically created** for Brazilian Portuguese. There is a language dependency.

WHO_AGENT and WHO_AFFECTED rules are only based on false positives, because the SRL system does not identify if an annotated segment represents an entity, person or organization (see Fig. 5). Based on the PALAVRAS's semantic tags, it is possible to determine if an SRL annotated segment has at less a token that represents an entity, PERSON or ORGANIZATION.

Rules for aspects WHEN, WHERE and WHY are based on false negatives and false positives. WHEN rules follow Baptista et al. [2] theory to identify temporal expressions (see Fig. 6). WHERE rules identify local expressions (see Fig. 7) and WHY rules identify causative expressions (see Fig. 8).

Finally, SCORE rules were created based on a few number of annotated sentences (10) and integrated to de SRL+Rules system. Thus, SCORE rules are limited to this minimum set of annotates sentences.

PERSON[4] = [H, HH, Hattr, Hbio, Hfam, Hideo, Hmyth, Hnat, Hprof, Hsick, Htit, hum]
ORGANIZATION = [admin, org, inst, media, party, suborg]
$\in$ = "is an element of"
$\notin$ = "is not an element of"

Rule 1: If a sentence has a segment annotated by the SRL system that contains a token associated to PALAVRAS's semantic tags PERSON or ORGANIZATION, and iff the token is not a REPENTINO's[5] local lexicon, then the segment will be correctly annotated as WHO_AGENT or WHO_AFFECTED. Otherwise, the annotation will be removed.

Input: *<aspect SRL=WHO_AGENT>The president</aspect> says that some constructions are already underway, and <aspect SRL="WHO_AGENT">others</aspect> will start soon.*

president_(Hprof) $\in$ PERSON $\notin$ local_lexicon

others(diff) $\notin$ PERSON

Output: *<aspect SRL=WHO_AGENT>The president</aspect> says that some constructions are already underway, and others will start soon logo.*

Fig. 5. WHO_AGENT/WHO_AFFECTED rules

[4] PALAVRAS's semantic tags for entities PERSON and ORGANIZATION.

[5] Named-entity lexicon created in HAREM-2005 (www.linguateca.pt/repentino/).

PREP = [de, em, a, por, para]
PRON = [ele(s), ela(s), este(s), esta(s), esse(s), essa(s), aquele(s), isto, isso, aquilo, aqui, aí, ali, outra(s)]
DET = [a(s), o(s), um, uns, uma, umas, à(s)]
day_lexicon = [segunda-feira, terça-feira, quarta-feira, quinta-feira, sexta-feira, sábado, domingo]
time_adverb_lexicon = [hoje, amanhã, ontem, anteontem, tarde, madrugada, noite, meia-noite, manhã...]
time_lexicon = [microssegundo, segundo, minuto, hora, dia, semana, mês, ano, década, milênio, época...]
"+/-" = follow_or_not

Rule 1: If a sentence has PREP + (PRON|DET) + time_adverb_lexicon + PREP + (PRON|DET) + day_lexicon +/- NUM, then the sentence is annotated as WHEN.

"A chuva complicava o trânsito na manhã desta segunda-feira, 16."

na_(PREP+PRON) + manhã_(time_adverb_lexicon) + desta_(PREP+PRON) + segunda-feira_(day_lexicon) + NUM

Rule 2: If a sentence has PREP + (PRON|DET) + day_lexicon, then the sentence is annotated as WHEN.

"Um terremoto atingiu Japão nesta segunda-feira matando 9 pessoas."

nesta_(PREP+PRON) + segunda-feira_(day_lexicon)

Rule 3: If a sentence has PREP + (PRON|DET) +/- (TOKEN|NUM) + time_lexicon, then the sentence is annotated as WHEN.

"Aos 18 minutos, Maicon fez o primeiro gol."

Aos_(PREP+DET) + 18_(NUM) + minutos_(time_lexicon)

Rule 4: If a sentence has PREP + (PRON|ARTG) + time_adverb_lexicon, then the sentence is annotated as WHEN.

"A quarta medida foi aprovada nesta madrugada."

nessa_(PREP+PRON) + madrugada_(time_adverb_lexicon)

Fig. 6. WHEN rules

Rule 1: If a sentence has a segment annotated by the SRL system that contains a PREPOSITION "em", followed or not by the DETERMINER/PRONOUN, followed by a NOUN other than a temporal expression, then the segment will be correctly annotated as WHERE. Otherwise, the annotation will be removed.

Input: *"<aspect SRL=WHERE>On Sunday</aspect>, a bloody battle took place."*

on_(PREPOSITION) + Sunday_(temporal expression)

Output: *"On Sunday, a bloody battle took place."*

Rule 2: If a sentence has "em" + uppercase expression, then the sentence is annotated as WHERE.

"O senador Marcos nasceu em São Paulo"

em_(PREPOSITION) + São Paulo_(uppercase expression)

Fig. 7. WHERE rules

cause_lexicon = [por isso, com isso, porque, devido a, por causa de, por força de, em função de, em virtude de, em razão de, em decorrência de, em consequência de, pois, visto que, já que, causado]

Rule 1: If a sentence has cause_lexicon, then the sentence is annotated as WHY.

"O senador teve seu estado de saúde piorado, por causa de complicações gastrointestinais."

por causa de_(cause_lexicon)

Rule 2: If a sentence has the PREPOSITION "por" + infinitive_verb, then sentence is annotated as WHY.

"Já Poliana Okimoto ficará fora de a decisão de os 800m livre por estar com infecção intestinal."

por_(PREPOSITION) + estar_(infinitive_verb)

Rule 3: If a sentence has expression "graças a" + DETERMINER, without being part of "dar graças" expression, then the sentence is annotated as WHY.

"Graças ao médico, o paciente sobreviveu."

Graças_a_(expression) + o_(DETERMINER)

Fig. 8. WHY rules

3.3 REMBRANDT System

The REMBRANDT system automatically annotates *microaspects* equivalent to the NE categories (*cf.* Table 4). It covers WHEN, WHERE, and SITUATION. Fig. 9 shows a sentence annotated using the REMBRANDT system. Note that the segment “Pan American Games” represents the entity EVENT, “Tuesday” represents the entity TIME, and “Maracazinho” represents the entity LOCAL.

The Brazilian volleyball team won over the Finnish team this <aspect EM="WHEN">Tuesday</aspects> by 3 sets to 0 in <aspect EM="WHERE"> Maracazinho</aspect>, on the <aspect EM="SITUATION">Pan American Games</aspect>.

Fig. 9. *Microaspect* sentence annotation using REMBRANDT system

3.4 Machine Learning Approach

Nowadays, computers have the capacity to automatically learn tasks based on experiences. These experiences are formed by a set of examples called "instances". In this work, the task to be learned is the "microaspect identification". With a manually annotated corpus the task follows a Machine Learning (ML) supervised paradigm, where the training set is formed by *instance-class* pairs called *labeled data*. Thus, the instances are represented by the sentences in the corpus, and the classes are represented by the annotated aspects in the sentences. For that reason, we proposed to use ML techniques to create a *microaspect* classifier.

Microaspect identification is a multi-label classification problem. In this work we apply the “problem transformation methods”, which aims to transform the multi-label classification problem into a set of binary classification problems. Therefore, many classifiers were created. We only chose the best classifier for each *microaspect*. In total we obtain 8 binary classifiers (see Table 2).

In total, six types of features provided by PALAVRAS *flat* format were defined: bag of words, lemmas, POS (part-of-speech), semantic tags, lemmas+POS and POS+semantic-tags. Each feature is represented by unigrams “(1, 1)”, bigrams “(2, 2)” and bigrams+trigrams “(2, 3)”. The result of the representation of the six types of features generates many classifiers for each *microaspect*. For example, the classifier "(2, 3) POS" was created based on all bigrams and trigrams of the part-of-speech of all the words in the corpus. The total number of created classifiers is 144.

We use the SVM (Support Vector Machine) supervised algorithm to classify *microaspects* in a sentential level. This algorithm is the most used in NLP classification tasks because, it is the best to deal with big dimensional space vectors.

4 Experiments and Results

The evaluation was measured by 4 metrics: R (Recall) – percentage of actually positive instances that were labeled as such; P (Precision) – percentage of instances labeled as positive that actually belong to this class; F1 (F1-score) – harmonic mean of the P and R; A (Accuracy) – total number of hits over the total number of instances.

All systems were tested on a set of sentences manually annotated with aspects from the CSTNews corpus. In addition, we tested the combination of the systems (SRL+REMBRANDT, SRL+Rules+REMBRANDT) only on the aspects WHEN and WHERE. In total, there are 322 annotated sentences. Table 5 shows the results of the best systems. The best result was obtained by the SRL+Rules for SCORE (F1=1.000) for the class "YES", whereas all the SCORE rules were created only for a few numbers of false positive sentences. In a majority of cases, the best results were obtained by SRL+Rules system. That proves that the handcrafted rules improved the performance of the SRL system. The worst result was found in the SRL system for HOW (F1=0.040). That happens because the SRL system considerably failed (many incorrect annotated sentences), and, in some cases, human annotators failed. REMBRANDT is the only system that can identify SITUATION. Note that all results of both, F1 of the class "NO" and the accuracy (A), are the highest.

Table 5. Best results using system approach

		"YES" class			"NO" class			A
Microaspect	**System**	**R**	**P**	**F1**	**R**	**P**	**F1**	
WHO_AGENT	SRL+Rules	0.592	0.664	0.626	0.797	0.743	0.769	0.624
WHO_AFFECTED	SRL+Rules	0.417	0.368	0.391	0.836	0.862	0.849	0.758
WHEN	SRL+Rules	0.947	0.504	0.657	0.717	0.978	0.827	0.770
WHERE	SRL+Rules	0.804	0.474	0.596	0.812	0.952	0.876	0.811
WHY	SRL+Rules	0.469	0.789	0.588	0.986	0.944	0.966	0.935
HOW	SRL	0.111	0.024	0.040	0.872	0.972	0.919	0.851
SITUATION	REMBRANDT	0.231	0.750	0.353	0.993	0.933	0.962	0.929
SCORE	SRL+Rules	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Differently from the previous systems, the ML classifier was trained and tested with all the 322 sentences. We used a **stratified** strategy on the corpus to ensure the same proportion of classes in each subset. The corpus was ten times stratified, where the sentences were divided into 70% for training (225 instances) and 30% for testing (97 instances), for each iteration. Thus, the final result is the average value of the iterations. Table 6 shows the best classifier for each *microaspect*. The best result was obtained by the "(1, 1) semantic" classifier for WHEN (F1=0.615). That occurs due to the fact that the PALAVRAS semantic tags contain a time lexicon. The worst result was obtained for microaspect SCORE (F1=0.000). This happens because CSTNews has few sentences annotated with SCORE. Note that most of the best classifiers are represented by unigrams "(1, 1)". Finally, in the majority of the cases the classifier "(2, 3) POS+semantic" obtained the best results because it have more linguistic knowledge (part-of-speech and semantic) than the others classifiers.

5 Conclusions

In this work we proposed a method to automatically identify *microaspects*. *Microaspects* represent local segments that form a sentence. Firstly, we evaluate a system approach based on semantic roles, named-entity categories, handcrafted rules and a combination of them, using de CSTNews corpus. The results proves that the SRL+Rules system obtained the best result in the majority of the cases. That means that handcrafted rules improved the SRL system performance. However, there were some problems in the identification process: the SRL system failed to correctly identify some sentences, affecting the performance of SRL+Rules system, and the PALAVRAS parser failed to analyze some sentences.

Table 6. Best classifiers using the Machine Learning approach

Microaspect	Classifier	R	P	F1	A
WHO_AGENT	(2,3) POS+semantic	0.538	0.636	0.583	0.691
WHO_AFFECTED	(1, 1) lemmas	0.222	1.000	0.364	0.854
WHEN	(1, 1) semantic	0.522	0.750	0.615	0.845
WHERE	(2, 3) POS+semantic	0.471	0.615	0.533	0.856
WHY	(2, 3) POS+semantic	0.200	0.500	0.286	0.897
HOW	(1, 1) bag_of_words	0.250	1.000	0.400	0.938
SITUATION	(1, 1) lemmas+POS	0.333	1.000	0.500	0.959
SCORE	All	0.000	0.000	0.000	0.000

Secondly, we evaluate a ML approach based on lexical, part-of-speech and semantic features. The results are not satisfactory because of the few number of annotated sentences. We believe that ML results could be better with more annotated instances in the corpus.

We can not compare the two approaches, because both approaches (system and ML) were tested in different ways. It is important to say that the subjectivity present in the process of the corpus manual annotation could affect the performance of the two approaches.

In conclusion, the system approach proves that *microaspects* can be automatically identified for Portuguese news texts. The main goal to identify aspects is to automate the Summarization task and to assist other NLP tasks (e.g., Question-Answering). This is a *novel* work for Brazilian Portuguese. Finally, we created a baseline to compare results with future systems, e.g. bag of words (unigramas).

Future work will focus on creating a *macroaspect* classifier. Our aim is to build automatic classifiers that cover all aspects defined in the CSTNews corpus (previously defined by TAC).

Acknowledgments. Part of the results presented in this paper were obtained through research on a project titled “Semantic Processing of Texts in Brazilian Portuguese”, sponsored by Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91.

References

1. Alva-Manchego, F.: Anotação Automática Semissupervisionada de Papéis Smânticos para o Português do Brasil. Dissertação, Instituto de Ciências Matemáticas e de Computação - ICMC-USP (2013)

2. Baptista, J., Hagège, C. and Mamede, N: Capítulo 2: Identificação, Classificação e Normalização de Expressões Temporais do Português: A Experiência do Segundo HAREM e o Futuro. In: C. Mota e D. Santos (eds.), Desafios Na Avaliação Conjunta do Reconhecimento de Entidades Mencionadas: O Segundo HAREM. Linguateca (2008)
3. Barrera, A., Verma, R., and Vincent, R.: SemQuest: University of Houston's Semantics-based Question Answering System. In: Proceedings of the Fourth Text Analysis Conference, Maryland, USA. National Institute of Standards and Technology. (2011)
4. Bick, E.: The Parsing System “Palavras”: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Ph.D. thesis, University of Arhus. Denmark (2000)
5. Cardoso N: REMBRANDT - Reconhecimento de Entidades Mencionadas Baseado em Relações e Análise Detalhada do Texto. In: C. Mota & D. Santos (eds.), Desafios na Avaliação Conjunta do Reconhecimento de Entidades Mencionadas: O Segundo HAREM. Linguateca, 2008, pp. 195-211 (2008)
6. Cardoso, P., Maziero, E., Castro Jorge, M., Seno, E., Di Felippo, A., Rino, L., Nunes, M., and Pardo, T.: A Discourse Annotated Corpus for Single and Multi-document Summarization of News Texts in Brazilian Portuguese. In: Proceedings of the 3rd RST. Brazilian Meeting, pp. 88-105 (2011)
7. Genest, P., Lapalme, G., e Yousfi-Monod, M.: HEXTAC: the Creation of a Manual Extractive Run. In: Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology (2009)
8. Li, P., Wang, Y., Gao, W., and Jiang, J.: Generating Aspect-oriented Multidocument Summarization with Event-aspect Model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pp. 1137-1146, Stroudsburg, PA, USA. Association for Computational Linguistics (2011)
9. Makino, T., Takamura, H., and Okumura, M.: Balanced Coverage of Aspects for Text Summarization. In: Proceedings of the Fourth Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology (2011)
10. Owczarzak, K. and Dang, H.: Who wrote What Where: Analyzing the Content of Human and Automatic Summaries. In: Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages, pp. 25-32, Portland, Oregon. Association for Computational Linguistics (2011)
11. Palmer, M., Gildea, D., e Xue, N.: Semantic Role Labeling. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers (2010)
12. Rassi, A., Zacarias, A., Maziero, E., Souza, J., Castro, L., Balage, P., Cardoso, P., Camargo, R., Agostini, V., Filippo, A., Seno, E., Rino, L., and Pardo, T.: Anotação de Aspectos Textuais em Sumários do Córpus CSTNews. Relatório Técnico NILC TR-13-01, Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (2013)
13. Steinberger, J., Tanev, H., Kabadjov, M., and Steinberger, R.: JRC's Participation in the Guided Summarization Task at TAC 2010. In Proceedings of the Text Analysis Conference, pp. 1-12 (2010)
14. Swales, J. Genre Analysis: English in Academic and Research Settings. Cambridge University Press, Cambridge, UK (1999)
15. White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., and Wagsta, K.: Multidocument Summarization via Information Extraction. In: Proceedings of the First International Conference on Human Language Technology Research, HLT '01, pp. 1-7, Stroudsburg, PA, USA. Association for Computational Linguistics (2001)

A Logistic Regression Model of Irony Detection in Chinese Internet Texts

Frank Z. Xing and Yang Xu

Department of Information Management, Peking University, Beijing, China
{xzt,yang.xu}@pku.edu.cn

Abstract. The research of sentiment analysis has become fascinating with the support of emerging Internet language material. In this paper, irony in Chinese is investigated as a sentiment that has not been meticulously studied. We describe here a set of features and their computational formalization for detecting irony at a linguistic level. Comments from online forum are collected and detected whether ironical or not, with a logistic regression model. The efficacy and validity of our model is proved by comparison with other popular learning methods and statistical testing. The model achieves a performance close to state-of-the-art results in English and Italian from recall and accuracy perspective.

Keywords: sentiment analysis, formalized features, intentional meaning, logistic regression

1 Introduction

Irony is an ubiquitous phenomenon in many natural languages. There has been a long-time discussion in English rhetorics about irony as a figure of speech. In the discussion the classification of irony and the widely accepted definition of irony as “saying the opposite of what is meant” are proposed [1]. With the rapid development of Internet content, the task of irony detection gains significance in both sentiment analysis and its practice, such as opinion mining. However, the topic of irony has received little serious computational treatment in the previous discussion. Thus, works emphasized on sentiment analysis start to employ the method of summarising and elaborating linguistic features of irony for computational formalization [2–4].

However, most of previous studies are dominated by irony detection in English contexts, other languages have been neglected for a long time. Although [5, 6] starts to discuss automated irony detection in Italian and Brazilian Portuguese, there is still a lack of literature dealing with non Indo-European languages. While irony is a pervasive linguistic phenomenon, some of the specific features vary with cultural and structural properties of language. Transplanting all the features of irony in English to other languages as a whole will meet technical difficulties, and the performance of the approach will be weakened. To solve this problem, this paper, accordingly aims to explore characteristic features of irony in Chinese contexts, and improve the detection techniques.

pp. 239–249; rec. 2015-01-20; acc. 2015-02-27

The approach to detect irony in Chinese goes in three steps. First, theories on irony are systematically reviewed to clarify the main features of irony, and their applicability in Chinese are validated. Second, key features are summarized and utilized for computational formalisation to form independent variables. Third, a logistic regression model with threshold is employed to synthesize the features for a better-performanced solution of irony detection. Since the automatic detection of irony in Chinese has not been well studied, we take the state-of-the-art results in English, Portuguese and Italian for comparison. The approach provides new insights on computational formalisation of irony. Meanwhile, through the evaluation part, some susceptible hypotheses are tested, hence our knowledge on Chinese irony from a computational linguistic perspective is extended.

The rest of this paper is organized as follows. Section 2 introduces related works on irony theories and and illustrates typical examples of irony sentences in Chinese. Section 3 describes the data we extracted from Internet forums and the formalisation of features in detail. The performance of the model is discussed and evaluated in Section 4, highlighting its validity. Section 5 is the conclusion.

2 Irony in Chinese contexts

Theorist [7] suggests that identification of irony, even in face-to-face communication can be difficult due to perception of irony varying among peoples. Many people realize irony while others not, because of the ambiguity of language. The difficulties in irony perception of human indicate the complexity of computer based irony detection.

Irony can be understood from three perspectives, as a figure of speech, as mention [3] and as phrasal pattern [8]. Traditionally, irony as a figure of speech, has been mainly characterised as negation that conveys the original meanings of utterance. However, Grice in [9] finds out violations as another feature in irony. He defines irony as “a case of flouting the cooperative principle, by violating the maxim of quality”. Giora in [10] elaborates Grice’s violations into indirect negation and graded salience. The former one refers to the linkage between the literal and implied meanings are indirect but negative. However, the latter one argues that irony text can be understood polysemously whereas irony occurs when salience meanings have been processed by speakers over other (mainly literal) meanings. A further development of this theory is to understand irony as an indirect speech act [8]. Such theory regards irony as an intentional expression of insincerity that cannot be limited to assertions but can also be applied to many other occasions such as congratulations.

Another theory about irony concerns contextual-appropriateness, which points out that rather than violating maxim of quality, irony shall be defined as mention where utterance seems to be literally not appropriate to its context [3]. Except for unintentional irony, ironical utterance can be inappropriate but relevant to its contexts. The speaker intentionally knows such inappropriateness while he/she intends the audiences are also aware of these as well. Irony is not necessarily negative, but utterance with inappropriateness from this angle. This theory extends the boundary of irony and suggests that understanding of irony requires two steps that includes (1) understanding the intentional meaning of irony, which is probably through graded salience and (2) assessing the semantic conflicts.

Partington in [11] also mentions irony patterns in phrase usage, mainly by combining elements within the phrase of opposing evaluative polarity. This is named phrasal irony. However, in Chinese the concept of “phrase” is more free than in English. Thus we make extend the range of polarity to a whole sentence.

To conclude, theories based on English contexts have provided a relatively comprehensive map of irony. Key descriptive features include (1) relevant but inappropriate to its contexts, (2) intentionally or unintentionally perceived inappropriateness and (3) with certain phrasal patterns. Reyes et al. in [12, 13] proposed corresponding formalizable dimensions called *Signatures*, *Unexpectedness* and *Emotional scenarios* and the sub-branches.

The dimensions are basically applicable to irony in Chinese. However, features can be different. For example, Reyes et al. in [13] suggested capitalized words as a hint of pointness, but this phenomenon does not exist for Chinese characters. Emoticons and punctuation marks are more appropriate for emotional scenarios. Thus in Chinese, *signatures* mainly consists of counter-factuality in [14]. Example 1 in Table 1[1] illustrates the “clever” does not carry its literal meaning, or he would not be uncomfortable with his father’s words and parental love. A general idea of irony in Chinese is that, irony is presented as text whose signifying is different from its signified. In other words, the literal meaning of the text is different from the real semantic meaning the speaker intended to express. Unexpectedness suggest that the collocation is not common. In Chinese the collocation can be sentence level inappropriate. For instance in Example 2, Chinese cruller can not be normally modified by adjectives like “slick” and “sly”.

Table 1. Typical Examples of Irony in Chinese

Example 1	我那时真是聪明过分，总觉他说话不大漂亮。 I was too clever at that time, as I never thought he spoke appropriately.
Example 2	我不欺不瞒地说，他是个油条，八面见光，是个玲珑灯。 I will not cheat you, he is a Chinese cruller, slick and sly.
Example 3	地沟油也是民生工程之一啊！ Gutter oil is also one of the service offered to the public!

Chinese scholars [15, 16] also point out that irony usually appears in contexts with certain emotion. For instance, Example 1 shows an emotion of regretting and self-accusation. Pragmatically speaking, the contraction in meanings is usu-

[1] Example 1 is from Chinese essayist Zhu Ziqing’s famous prose “My Father’s Back”; Example 2 is from Zhu Chunyu’s novel “In a Sea of People”; Example 3 is in our dataset extracted from the Internet.

ally used to express certain emotions, to help better highlight some traits of the objects. Therefore, tracing emotional fluctuation in certain contexts can be a good indicator of expression of irony. In Chinese, this emotional scenario is dominated by coexistence of commendatory and derogatory terms. There have been researches about polarity [17, 18]. Sentence level polarity well represent the emotional scenario.

3 Model

3.1 Dataset for Irony Detection

In consistent with previous research, the dataset used is extracted from the Internet. However, Twitter is not the most popular platform for disseminating and sharing information and opinions in Chinese. In order to make a better coverage and representativeness, the data comes from three popular Chinese news forums – Tencent News, Sina Social News and Tianya Forum.

From these news forums, we choose 15 topic articles from the most popular chart (ranked by page view) in September 2013 and September 2014, each of which is followed by thousands of feedback and comments. Because irony is not a frequent phenomenon in natural language, we filtered irrelevant comments to increase the odds of irony. Literature [19] suggests by incorrectly classifying all 2,795 ironic simile speeches as non-ironic, the system still achieves an overall F-measure of 0.81. Such a system has no irony detection capabilities at all, yet achieves reasonable performance just because of the imbalance of non-irony to irony among speeches.

As a result, 2,602 comments in total are extracted from bellowing these news articles. 416 comments are marked positive of irony according to the survey, which suggests an overall ratio of 16.0%. The character count for each comment ranges from 4 to 191, with a variance of 611.7. Therefore comment length is regularized in the following formalization.

3.2 Computational Formalization of Irony Features

A speech is defined as a stream of meaningful text that carries emotion and intention. The specific form of a speech in the experiment is a short online comment. The intentional meaning of a comment and the article it belongs to are related. Theme refers to the literal meaning of an article. The intentional meaning of a comment is assumed the same as the theme of its related article.

Take that U.S. president Barack Obama refuted claims that hostility to his presidency was due to racism, by noting that “I was actually black before the election” as an example [19]. This statement was both literally true and ironic. In this case the literal meaning can be concluded directly from his statement. However, if we do not know about the context, it would be extremely difficult to summarize the intentional meaning. In our experiment, we fortunately capture the intentional meaning by summarizing the article, which is highly probably about racism, indicating irony by the difference between literal meaning and intentional meaning.

Inspired by automatic indexing, we utilized the keyword vector of a text to represent its literal meaning. Then the vector angle between the keyword vector of a speech and its theme is a good metric of the deviation from the speech's intentional meaning to its literal meaning.

We use T_i to denote a certain theme shared by several speeches $S_1^i, S_2^i, ..., S_j^i$. i denotes the number of articles and j denotes the number of comments related to T_i.

$\alpha(T_i) = (w_1 K_1^{T_i}, w_2 K_2^{T_i}, ..., w_m K_m^{T_i})$ denotes the weighted keyword vector of theme T_i. $\alpha(S_j^i) = (w_1 K_1^{S_j^i}, w_2 K_2^{S_j^i}, ..., w_n K_n^{S_j^i})$ denotes the vector of speech S_j^i. K_i is the i^{th} keyword, w_i is the normalised weight of K_i.

Feature 1: $Re(T_i, S_j^i) = \cos < \alpha(T_i), \alpha(S_j^i) >$ measures the similarity of literal meaning and intentional meaning of S_j^i. Therefore $Re(T_i, S_j^i)$ is negatively related to the probability that S_j^i expresses irony. In the calculation, we regard $|\alpha(T_i)|$ as a constant because the theme apparently generates more keywords to make $|\alpha(T_i)|$ far more large than $|\alpha(S_j^i)|$. Consequently we are only concerned by the length of a speech.

$$Re(T_i, S_j^i) = \frac{\alpha(T_i) \cdot \alpha(S_j^i)}{|\alpha(T_i)||\alpha(S_j^i)|} = \frac{1}{|\alpha(S_j^i)|} \sum_{K_m^{T_i} = K_n^{S_j^i}} w_m^T w_n^S \tag{1}$$

Feature 2: If the word vector[2] of S_j^i has the form $(w_1^S, w_2^S, ..., w_q^{S_j^i})$, $I(S)$ can be calculated as the number of inappropriate collocation $I(S) = \mathcal{N}(w_k^S, w_{k+a}^S), k \in \{1, 2, ...q - a\}$. The window length a can be adjusted to balance the probability of missing and mistaken inappropriate collocation.

Feature 3: If each word is assigned a value to denote its polarity, the corresponding emotion vector can be written as $\epsilon(S_j^i) = (E_1^{S_j^i}, E_2^{S_j^i}, ..., E_q^{S_j^i})$. Let $E^+ = |\sum_{r=1}^{q} E_r^{S_j^i}|, \forall E_r^{S_j^i} > 0$; $E^- = |\sum_{r=1}^{q} E_r^{S_j^i}|, \forall E_r^{S_j^i} < 0$. If $E^+ E^- \neq 0, E(S_j^i) = \frac{E^+ + E^-}{1 + \rho|E^+ - E^-|}$, otherwise $E(S_j^i) = \sigma(E^+ + E^-)$, ρ and σ are coefficients for smoothing.

To achieve the computational formalisation aforementioned, several techniques are employed. First, the software package, MMSeg for Java lucene Chinese analyzer, is used to deal with Chinese word segmentation. The maximum

[2] note the difference between "word vector" and "keyword vector", which is ignoring the weight values in $\alpha(S_j^i)$, A keyword vector of S_j^i is obtained as a vector $(w_1^S, w_2^S, ..., w_p^S)$.

matching algorithm that implemented the conversion from S_j^i to $(w_1^S, w_2^S, ..., w_p^S)$ is introduced in [20]. Second, we implemented the keyword vector generation algorithm introduced in [21]. The algorithm first builds a graph of the annotated sentences, then do recursions to estimate the weight of each edge. At the final stage weights are assigned to the words and those words whose weight equals to zero are deleted. Third, we referred to the method of fundamental sentiment word polarity computation in [18] to modify the emotion dictionary we constructed, in which each word is quantitatively described with its emotional intention, part of speech and its synonyms[3]. The calculation of $E(S_j^i)$ is based on the modified dictionary.

Here is an example of how the formalisation of features works. In our dataset, there is a theme article titled “星巴克：在中国更贵因为顾客长时间滞留店内喝咖啡 (Starbucks: More expensive in China for clients stay several hours to drink coffee)”. The TextRank algorithm generates $\alpha(T)$ as $(1*$ China, $0.44*$ Starbucks, $0.41*$ America, $0.25*$ coffee, $0.23*$ café, .., $0.1*$ (TV station), $0.1*$ quite, $0.1*$ claim, $0.1*$ (raw material)). A speech of this theme, apparently annoyed by the story of Starbucks, is “Starbucks is just a symbol of American junk food culture”. $\alpha(S)$ is calculated as $(1*$ symbol, $1*$ Starbucks, $0.1*$ junk, $0.1*$ culture). Therefore $Re(T,S) = \frac{0.44\times1}{\sqrt{1+1+1+1}}, \epsilon(S) = (0, 0.1, 0.4, -1, 0.1, 0, 0)$. We empirically set the coefficients (ρ, σ, a) to $(0.22, 0.35, 2)$, then $E(S) = 1.47, I(S) = 2$.

4 Experiment

2,302 randomly separated speeches in the dataset are utilized as the training set to estimate the coefficient of each feature. The rest 300 speeches form the testing set. The speeches in training set are annotated as ironical or not, manually through a survey. All interviewees are native speakers of Chinese with good education backgrounds. The theme and speech are presented to three different people for judgement. If more than two people recognized a speech as ironical, it will be annotated as a speech of irony. For the testing set we presented them to five people and annotated it as ironical if three or more people recognized it.

Several statistical machine learning methods can be applied for irony detection. The output space is a binary sequence to determine whether a speech expresses irony or not. Intuitively we choose a logistic regression model to deal with the classification problem. Instead of investigating to each feature respectively as in [12], logistic regression model synthesizes effects of the features. More specifically, we adjust the cut-off point to find out a proper probability threshold to balance between precision and recall. We also compared the model with artificial neural network model and decision tree (C4.5) model. The performance on testing set is illustrated in Figure 1.

The model is implemented with IBM SPSS Statistics 20. $Re(S), E(S)$ and $I(S)$ are used as independent variables. Notably in order to increase the accu-

[3] We have referred to the Chinese Dictionary of Emotion, works of the Natural Language Processing and Computational Social Science Lab, Tsinghua University (http://nlp.csai.tsinghua.edu.cn/site2/index.php/zh/resources/13-v10).

racy of the model, speech length is a control variable in the model. Descriptive analysis of dependent and independent variables is illustrated as in Table 2 below. As illustrated in the Table, the three features are significantly relevant to the property whether a text is ironic or not. $Re(S)$ and $E(S)$ are negatively correlated. This phenomenon indicates that the deviation of intentional meaning from literal meaning usually occurs with strong emotion.

Table 2. Correlations of Independent and Control Variables

	Mean	Min	Max	S.D.	0	1	2	3
0 Annotation	0.18	0	1	0.306	1.000			
1 Speech Length	28.733	4	191	24.733	-0.85	1.000		
2 $Re(S)$	0.108	0	1.76	0.298	-0.162*	-0.119	1.000	
3 $E(S)$	0.52	0	3	0.824	0.159*	-0.081	-0.328*	1.000
4 $I(S)$	1.79	0	7	5.621	0.138*	0.378	0.042	0.139

4.1 Result and Discussion

Table 2 suggests that all independent variables have a significant impact on the regression. The accuracy of the detection depends on the selection of cut-off point that indicates the significant level. By comparing the outcome of the regression model and the manually annotated result of the testing set, we can draw a confusion matrix to calculate precision, recall and F-measure. Emphasizing on different aspect of precision or recall, F can be written as $\frac{(\beta^2+1)PR}{\beta^2 P+R}$. We take parameter $\beta = 0.5$ and $\beta = 1$ into consideration, set the cut-off point from 0.1 to 0.4 with an interval of 0.05. The performance of regression model is listed in Table 3.

Table 3. F-measure with different β and cut-off point

Cut-off	0.4	0.35	0.3	0.25	0.2	0.15	0.1
Recall	9.024%	20.109%	47.890%	61.117%	81.198%	89.402%	95.371%
Precision	47.301%	41.083%	39.771%	37.530%	33.119%	28.467%	27.737%
F (β=0.5)	25.591%	33.992%	41.166%	40.669%	37.567%	32.960%	32.321%
F (β=1)	7.578%	13.500%	21.727%	23.251%	23.524%	21.591%	21.487%

From the table we observe that the recall ratio varies violently when different cut-off point is chosen. However, the change of precision is more smooth. Therefore the cost to lift precision is high. When β=0.5, the F-measure suggests a set of cut-off point to 0.3. When β=1, the F-measure suggests a set of cut-off point to 0.2. Under the second circumstance, recall almost doubles, while the decrease of precision is not significant. Thus we set parameter β to 1.

4.2 Evaluation

Table 4. Omnibus test result

	Model 1	Model 2
Speech Length	-0.085*	-0.081**
	[0.07]	[0.009]
Re	–	-1.454***
	–	[0.311]
I	–	0.272**
	–	[0.095]
E	–	1.248***
	–	[0.001]
Constant	-1.671***	-1.650***
	[0.095]	[0.021]
Record Number #	2302	2302
Nagelkerken's R-sqaure	0.186	0.321
Omnibus chi-sqaure	0.149	17.624***

We employed statistical testing to verify the features investigated in our model did impact the efficiency of irony detection. In Omnibus test we compare the two models. Model 1 only includes control variable and constant whereas Model 2 is a model with additional independent variables to consider formalized features. Table 4 illustrates the details[4]. According to Table 4, the three features significantly improve the performance of Model 1.

Figure 1 demonstrates the performance of the model trained in Section 4 on the testing dataset. It is worth pointing out that results in Cinese, English, Italian and Portuguese are not strictly comparable because of the differences in datasets, ratio of ironic texts (in our work 16.0%, in [6] 51.89% and in [5] 12.5%), and testing methods. This may explain the high precision in Portuguese. The Logistic Regression model exceeds ANN and Decision Tree on all the three indicators. Generally speaking, on recall and accuracy our model performed almost as good as in English (the reported accuracy is 71.17% in English, we achieved 60.30% in our model; reported recall ratio in English is 62.17%, and our result outperforms it at 71.21%). If the purpose of detection focuses on computer assisted classification, accuacy should be considered in the first instance. But the precision of irony detection in Chinese is remarkably lower. This phenomenon

[4] Note: *** p<0.01, ** p<0.05, * p<0.10.

suggests that irony detection in Chinese is more challengeable and other forms of expression tend to be misunderstood as irony. In fact, however we adjust the cut-off point, the precision hardly reach 50%.

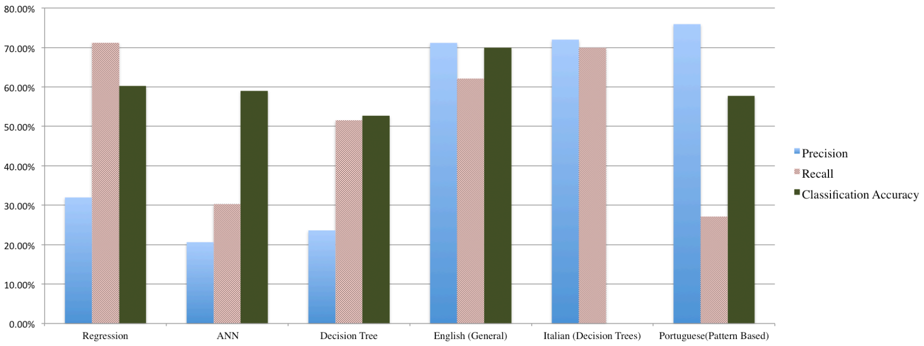


Fig. 1. Precision, Recall and Classification Accuracy of the testing set under different machine learning algorithms. The Logistic Regression, ANN and Decision Tree methods are carried on Chinese dataset. We compare our results with an overall evaluation of several methods in English [12] and results in Italian and Brazilian Portuguese. Accuracy in Italian is not reported.

There are reasons to hamper the irony detection result in Chinese to reach the standard of the other works in English, Italian and Portuguese. One of them could be the low quality of gold standard, which is generated by human annotators. The disagreement on irony is observed more frequently in Chinese during the annotation process. Another reason is the poor performance of natural language processing tools in Chinese. For example, there can be errors in the word segmentation step[5]. A certain word might have different emotional scenarios in different context. Literature [18] enumerates “骄傲 (pride:arrogance)” as an example. This word expresses positive opinion in certain contexts, while negative opinion in other circumstance. This ambiguity has a negative influence on the performance of the emotion dictionary because only the most frequent usage is taken into account.

5 Conclusion

In this paper, we introduce a logistic regression model for irony detection in Chinese online texts. Compared with irony detection in English, this model reinvestigates features of irony from previous research, and computationally formalized them to fit the different language environment. The significance of our features is proved by correlation analysis and Omnibus test. The performance of our model

[5] Based on this consideration, we have not discussed features involving part-of-speech.

comparing to similar works on English and other European languages suggests that precision of irony detection in Chinese still needs improvement.

Future work includes research on the detection framework and application of this model on analyzing public opinion.

Acknowledgements. This research is partly supported by National High Technology Research and Development Program of China (2012AA011101) and Research Fund of Undergraduate Research and Training Program of Peking University. The authors also want to thank the anonymous reviewers who gave them valuable comments on this work.

References

1. Feng, C: English Rhetorical Options. Foreign Language Teaching and Research Press, Beijing (2011)
2. Utsumi, Akira.: A Unified Theory of Irony and Its Computational Formalization. Volume 1: The 16th International Conference on Computational Linguistics. C 96–2162 (1996)
3. Attardo, S.: Irony as Relevant Inappropriateness. Journal of Pragmatics 32, 793–826 (2000)
4. Reyes, A., P. Rosso. Making Objective Decisions from Subjective Data: Detecting Irony in Customer Reviews. Decision Support Systems 53, 754–60 (2012a)
5. Francesco Barbieri, Francesco Ronzano, and Horacio Saggion.: Italian Irony Detection in Twitter: a First Approach. The First Italian Conference on Computational Linguistics CLiC-it 2014, pp 28–32 (2014)
6. de Freitas, Larissa A., et al.: Pathways for irony detection in tweets. Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM (2014)
7. Burgers, C., M. van Mulken and P. J. Schellens.: Type of Evaluation and Marking of Irony: The Role of Perceived Complexity and Comprehension. Journal of Pragmatics 44, 231–42 (2012a)
8. Partington, A.: Phrasal Irony: Its form, Function and Exploitation. Journal of Pragmatics 43,1786–800 (2011)
9. Grice, H. Paul.: Studies in the Way of Words. Harvard University Press, Cambridge, MA (1989)
10. Giora, Rachel.: On Irony and Negation. Discourse Processes 19, 239–264 (1995)
11. Partington, A.: Irony and Reversal of Evaluation. Journal of Pragmatics 39,1547–69 (2007)
12. Reyes, A., P. Rosso., T. Veale.: A Multidimensional Approach for Detecting Irony in Twitter. Language Resources and Evaluation 47, 239–68 (2013)
13. Reyes, A., P. Rosso.: On the Difficulty of Automatically Detecting Irony: Beyond a Simple Case of Negation. Knowledge and Information Systems 40(3), pp 595–614 (2014)
14. Reyes, A., P. Rosso., D. Buscaldi.: From Humour Recognition to Irony Detection: The Figurative Language of Social Media. Data & Knowledge Engineering 74, 1–12 (2012b)
15. Wen X.: On the Contextual Cues and Constraints of Ironic Utterances (in Chinese). Journal of Sichuan International Studies University. 17(6), 52–55 (2001)

16. Zhu, X.: A Study of Irony from a Pragmatic Point of View (in Chinese). Journal of Social Science of Hunan Normal University. 31(3), 128–129 (2002)
17. Wang, M., Shi, H.: Research on sentiment analysis technology and polarity computation of sentiment words. In: 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 1, pp. 331–334. IEEE (2010)
18. Li, R., et al.: A Mothod of Polarity Computation of Chinese Sentiment Words Based on Gaussian Distribution. Computational Linguistics and Intelligent Text Processing CICLing 2014, Part II, pp 53–61 (2014)
19. Yanfen Hao and Tony Veale.: An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes. Minds and Machines 20(4), 635-650 (2010)
20. Xu, L., Zhang, Q., Wang, D. D. and Zhang, J.: Research of Chinese Segmentation Based on MMSeg and Double Array TRIE. Advanced Research on Automation, Communication, Architectonics and Materials, 225-226(1-2), 945–948 (2011)
21. Cruz, F., Troyano, J. A. and Enriquez, F.: Supervised TextRank. Advances in Natural Language Processing, Proceedings, 4139, 632–639 (2006)

Finding Potential News from Trends Originating in the Blogosphere

Nigel Dewdney

University of Sheffield
Department of Computer Science
Sheffield. S1 4DP
acp08njd@sheffield.ac.uk

Abstract. Tracking current population interests by trends in online media of entities and topics has become increasingly popular. But while notable world events often spur online public discussion, some have been observed originating in social media postings. A natural question arises: Can analysis of social media trends be used to find mainstream newsworthy material? The work reported here takes initial steps towards answering this by investigating whether some characteristics of trending nouns and entities originating in blogs could predict a subsequent trend in mainstream news. Results show that many trends do originate in blogs, with approximately 12% seen subsequently in news media. Frequency based ranking provides a basis for selecting the most predictive trends. The study also suggests that named entity mentions, and co-occurrences thereof, may provide more focused trends than common nouns.

1 Introduction

The internet contains a wealth of information that is growing and changing all the time. Increasingly, information is not only provided by professional organisations via dedicated websites, but also the public at large via forums, newsgroups, social networking pages and web logs, or "blogs".. Individuals constantly add to the information (and possibly mis-information) on the web. Often these blogs reflect current circumstances for the authors. Could this wealth of material provide *new* interesting news worthy information?

Possession of the most up to date information can be of key commercial or strategic advantage in negotiation and decision making: "Knowledge is power" as the old saying goes. Knowledge of developing situations before they are widely known would therefore be of key advantage. There have in recent years been several well publicised occasions where social media has essentially broken news stories before the mainstream media. News of the Haitian earthquake of 2010 was first alerted to the world at large by messages on Twitter, and more recently the role of social media in supporting the "Arab Spring" popular uprisings are both examples where key indications were available online before being made available by the main stream news agencies. It would be desirable, therefore, to have an automated method of monitoring the social media for the unexpected.

pp. 251–263; rec. 2015-01-19; acc. 2015-02-25

The tracking of popular entities and topics in online media has become an increasingly common way to gain an insight into what populations are concerned about or interested in at any particular time. However, trends may result from current and recent events reported by the mainstream media outlets, or be of no significance to the wider world. We may assume that stories that have been published by the main stream media have been judged to be of interest. Ideally one would wish for new trending topics to be identified that are *not* linked to current or recent news stories. The requirement is for emerging topics to be identified that the user is not already aware of or can easily find in mainstream news outlets.

A step towards being able to find novel interesting information would be to investigate which trends in social media subsequently get picked up by the mainstream: how prevalent are they and could they be distinguished?

This paper reports on an analysis of trends in social media that do not have concurrent trends or coverage in the mainstream press . No attempt is made at this stage to characterise topics that trends relate to, or which will be picked up by news media. Neither does the study seek to quantify the proportion of news that occurs first in social media (others have examined this latter aspect). The aim here is to establish whether a significant proportion of trending features originating from social media could be used in news material selection or prediction, and whether there is advantage in using named entities.

The rest of the paper is organised as follows: section 2 summarises recent relevant and related work; section 3 provides a description of the data and the method of analysis employed; section 4 describes the analysis of the results; section 5 examines feature specificity through bi-grams of trends; finally section 6 gives conclusions and outlines future work.

2 Related work

Traditionally news story production has been dominated by professional sources and research focussed on news detection and tracking. For example see the TREC evaluations that ran a number of novelty detection tracks [1].

A popular approach has been to look for bursts of activity as indications of emerging interest in a topic: Kleinberg [2] in looking for time gaps in term occurrence in email data found bursts in topics coincided with interest to the author; bursts of linking activity have been observed by Kumar et al. [3] in the evolution of the "Blogosphere"; Gabrilovich et al. [4] have investigated the applicability of several distance metrics in finding novel information; Franco and Kawai [5] have investigated two approaches to detecting emerging news in blogs, by measuring linking evolution and by clustering the content of postings. Ha-Thuc and Srinivasan [6] have investigated using a log-likelihood estimate of an event within a topic model as an intensity metric; and Glance et al. [7] have examined bursts in phrases, mentions of people, and hyperlinks in blogs given a background of blogs published in the preceding two weeks. They hypothesise that product mentions in blogs may have predictive power for product success.

More recently work on emerging topic and trend detection has focussed on data from the micro-blogging web service Twitter, in which messages are restricted to 140 characters, and has been likened to chat rather than publication by Alvanaki et al. [8]. In their system, named “En Blogue”, they detect emerging topics by considering pairs of tags (augmented by extracted entities) at least one of which is frequent. Twitter provides its own proprietary trending topics service, but others have sought to provide similar functionality, e.g. [9], [10], and Benhardus [11] has compared different term weighting methods in Twitter trend detection.

The use of social media to predict future trends would seem to be a natural area for investigation given the establishment of evolving trends therein. Predicting the future from social media has seen interest in sectors such as movie commercial success - for example [12] - and political success - Tumasjan et al. [13] have investigated whether trends in party mentions can predict election outcomes. However prediction of topic popularity beyond news stories has not seen much investigation.

Research has also looked trend evolution in social media and how content spreads: Cha et al. [14] have studied the propagation of media content through the Blogosphere social network; Simmons et al. [15] have examined how quoted text changes as it is communicated; Clough et al. [16] have investigated whether a news story dependence on news agency text can be measured; while Asur et al. [17] have examined how trends persist and decay through social media, noting that many originate from providers such as CNN.

Evidence that social media content could pre-empt publication in the mainstream began to emerge in 2006. Lloyd et al. [18] in comparing the most popular named entities in news and blogs on a mentions-per-day basis found a small percentage of topics discussed in blogs existed before corresponding news-stories were published. A similar two-stream method to examine the characteristics of named entity trends and nouns that originated in blogs is followed here. In a different approach Leskovec et al. [19] looked at the evolution of “memes”, or short phrases, Although the majority of quotation was found in blogs, around 3.5% “meme” transfer was from blog entries to news, indicating social media origins. It is the type of material found to occur by these studies that is of interest here.

3 Data and analytic approach

Topics are often about tangible (named) entities: Azzam et al. [20] suggested that a document be about a central entity, and there is evidence that names can be effective in information retrieval tasks [21], [22]. This work investigates nouns and four types of entity as potentially trending features.

For data, the ICWSM-2009 [23], is used. The dataset, provided by Spinn3r.com, is a set of 44 million blog posts and news stories made between August 1st and October 1st, 2008. The data was pre-processed for the experiments here: Blog posts that have been classified either as “MAINSTREAM NEWS” or “WEBLOG” are extracted, while all others are discarded. Posts are not reliably language tagged,: no language filtering is applied, thereby maximising recall, which may also result in cross-language name reference inclusion. English part-of-speech tagging and named entity recognition are applied using the Stanford CoreNLP toolset without any modification [24] [25]. Data is grouped by day.

A standard Poisson model is employed for each feature frequency: This assumes that features occur at random and independently, the intervals between occurrences being Poisson distributed. The reciprocal of the expected interval gives the expected frequency. If a random variable X has a Poisson distribution with expectation $E[X] = \lambda$ then

$$P(X = k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, k \geq 0 \tag{1}$$

The mean frequency is simply the inverse of the expected gap between occurrences for the feature k, $1/\lambda$. The variance of the Poisson distribution is also λ. A trend is detected as a significant positive deviation from the expected distribution, its strength measured as the number of standard deviations in the associated gap reduction. For feature k with expected frequency $\frac{1}{\lambda_k}$ and observed frequency $\frac{1}{\lambda'_k}$, the strength of a trend in k is given by:

$$T(k) = \frac{\lambda_k - \lambda'_k}{\sqrt{\lambda_k}} \tag{2}$$

Frequencies for features are observed over a day. The daily trend in feature occurrence is measured in standard deviations given by $T(k)$ from the expected frequency. Expected frequencies are estimated by averaging those observed over preceding days. This does require a certain amount of "burn-in" time to establish a reasonable estimate of the average frequency $1/\lambda_X$ for each feature $X = 1, 2, ...$ Feature frequencies are calculated, therefore, on a daily basis while average frequencies are calculated on an accumulative one, i.e. no "window" is applied. (In a larger study a rolling interval would be more appropriate to account for long term drifts in language use.) Counts are calculated for each feature in each media category and Laplacian smoothing applied to account for unseen (new) features.

We may expect small changes in occurrences to be more significant for features that do not occur very often than for high frequency features. The use of Poisson models parameterised by feature counts for each stream affords relative trend measurements, so that deviations in feature frequencies that have different means in each media stream can be compared. Similarly, deviations in the frequencies of different features can be compared.

4 Trend Analysis

For each experiment a bedding-in time of seven days was arbitrarily chosen. Thereafter feature counts were calculated for each day and compared to the rolling average, reporting at the start of the next day. On any one day in the experimental period the top trending features by deviation from their average daily frequency to date are selected subject to a minimum of ten standard deviations above the average, and having more than five occurrences on the day. These thresholds were chosen to minimise trending due to movement caused by natural variation and to have some resilience to poor frequency estimation for very rare features. Any feature trending in the news is tagged and is not considered for the following seven days. If a news-trended feature does not trend in the news for a period of seven days it is then re-established as potentially blog-trending.

Feature types examined are those tagged as Nouns, Person names, Place names, Organisation names and Miscellaneous entities (others of undermined type). The number of trending features that were unique to blogs in the period and the number that subsequently trended in the main stream news media are shown in Table 1. Features may have trended more than once during the period but are only counted once.

Table 1. No. of trends originating in blogs & subsequently trending in news

Type	Trenders	News post trend	%
Nouns	9450	1741	18.4%
Misc	4350	221	5.1%
Location	4650	571	12.3%
Person	5450	740	13.6%
Organisation	5250	589	11.2%
Totals	29150	3862	13.2%

Blog trending features may not necessarily trend subsequently in the news, or even appear at all. Ranking by trend strength would seem to be a natural choice for prioritising and selecting trends as likely to trend subsequently in the news. However, as shown in Figure 1 graphs (a..d), having separated trending features into classes of those that subsequently trend in news, those that appear without trending in news, and those that appear uniquely in blog vocabulary, one finds that ranking by trend strength , measured in std. deviations, favours vocabulary that is unique to blogs. Ranking by feature frequency on the day of the trend, however, favours vocabulary that appears in the news, including that which subsequently trends. Note that ranking by frequency does not yield a proportion of subsequently trending features that is proportional to the number of ranked features. This is more marked for nouns than named entities.

Graphs (e) and (f) in Figure 1 show the proportions of feature type broken out from subsequently trending features and blog-unique features respectively. Miscellaneous entities are most likely to be unique to blogs, while Locations are the most likely to trend subsequently in news for the highest ranking features. However, overall, subsequent news-trending nouns are more likely to be selected than any particular entity type while being least likely to be a feature unique to blogs.

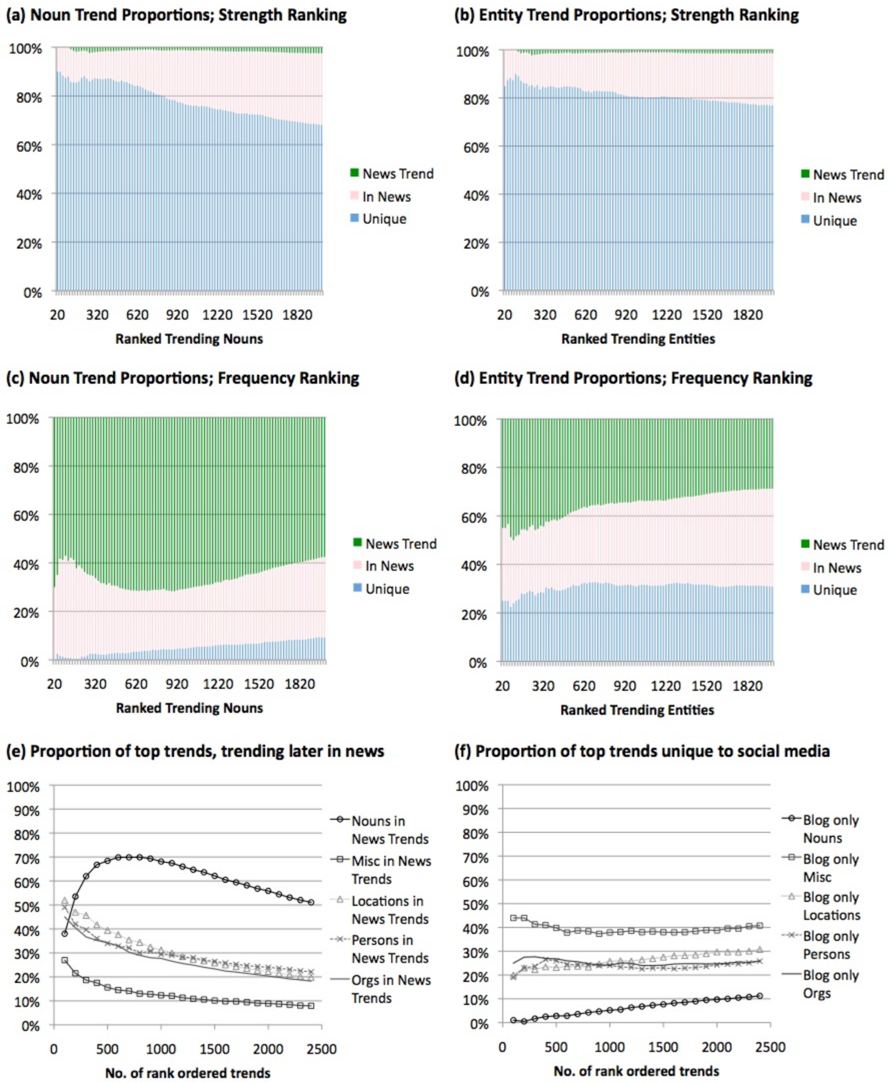


Fig. 1. Proportions of feature types in future news trends when ranked: (a) Nouns by trend strength, (b) Entities by trend strength, (c) Nouns by raw count, (d) Entities by raw count. (e) Prediction of news trend precision with raw count ranking. (f) Blog-unique features with raw count ranking.

For each feature type the top fifty blogs trends, having not trended in the previous seven days of news stories, were further examined. Note that this represents only a minority of all trends in social media, but the most significant for the purposes here. Although only the most significant trends originating in social media are examined, one should not expect all trends to be reflected in subsequent news stories; they may simply not reflect news-worthy material, or have been overlooked by media organisations. This is indeed the case with on average 38% of blog trending features (46% of top fifty by type) seen subsequently trending in news.

The number of mentions of selected terms in weblogs per day is greater than that in news stories. However, as the counts have not been normalised for number of posts, this should not be surprising as the number of blog posts is much higher than mainstream media articles (by about 20:1 for this corpus). Blog posts mentioning top trending nouns are particularly higher in number than the number of corresponding news articles, while generally less so for named entities An explanation for this is could be the higher number in average use of nouns compared with particular named entity mentions indicating a large background use of the nouns in question. This would suggest that trending named entities are more topically specific than trending nouns.

Figure 2 shows the evolution of the top two trends for each of the feature classes. In these graphs, only positive trending behaviour is shown. The trend strength is measured in number of standard deviations from the expected value. Trend activity in news is plotted on the negative y-axis (i.e. trend strength × -1). A solid line on the plot indicates when the trend behaviour in the blog is valid, i.e. not filtered from a previous or current news trend. Note that with even within the top selections illustrated one can observe a range in feature trend patterns. It may be possible to identify some patterns for features that would allow elimination or promotion (e.g. periodic trending). This is left for future work.

Just because a feature may trend in social media and then in the news does not mean that they are topically linked. It is well known that some features, be they generic nouns or named entities, are more specific in what they refer to than others. Unsurprisingly many topics can be found in posts giving rise to trends in generic nouns seen here. For example posts featuring “design” on 9th August include such diverse topics as the Olympics, a design conference, website design, guild badges, and peta-scale computing to name just a few. Subsequent news stories for “design” are similarly diverse, covering jobs, product reviews, etc.

Named entities seem little better. For example, trending location entities often result from a higher than average number of posts that refer to different events from those places. For example, “Mass.” arises from many posts about activities from multiple locations within Massachusetts. Subsequent news stories also have a wide spread of topics. However, some topics referring to a location are related to later news stories: “Pune” trends from posts of multiple topics including rain storms, Indian celebrities, and personal travel, but also commentary on a gang rape and murder that had occurred there. The subsequent news story is about lack of progress in the case.

Some named entities are more specific, though, and therefor topically more predictive. “Fannie” refers to the U.S. mortgage company Fannie Mae. The economic crash of 2008 had just got underway when ICWSM 2009 corpus was

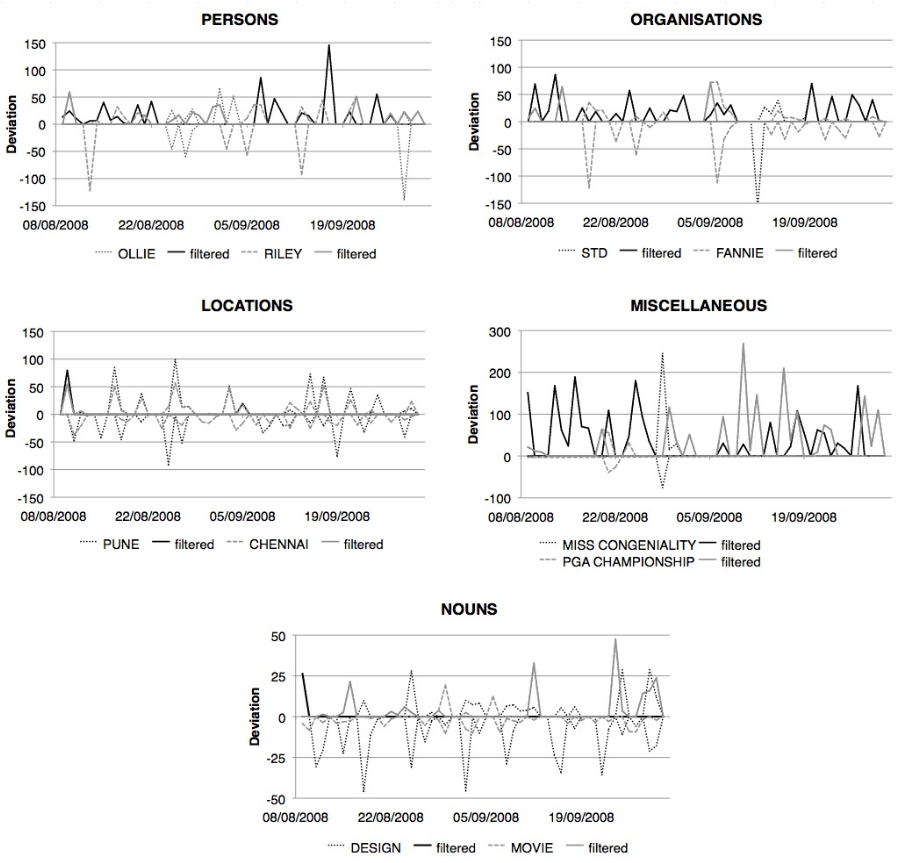


Fig. 2. Trend history for top two nouns and entity types trending in blogs prior to news. Trends (positive deviation from expected) in blogs plotted as positive deviation, and in news plotted as negative deviation.

collected, and there was much speculation about whether Fannie Mae (and its counterpart Freddie Mac) would need a U.S. government bailout. The news story trend occurred as various U.S. political figures reacted to market concerns, calling for appropriate aid, the following headline from the Chicago tribune being typical:

> “U.S. plans mortgage bailout”
>
> - http://chicagotribune.com/business/

Another example where an entity name was sufficiently specific to a topic is “PGA Championship” which referred to the 90^{th} PGA golf championship. Blogs centred on discussion surrounding players ahead of the playoffs as in:

> “Jack Nicklaus Isn’t Sure if Sergio Garcia Will Ever Win a Major”
>
> - http://golf.fanhouse.com/2008/08/20/

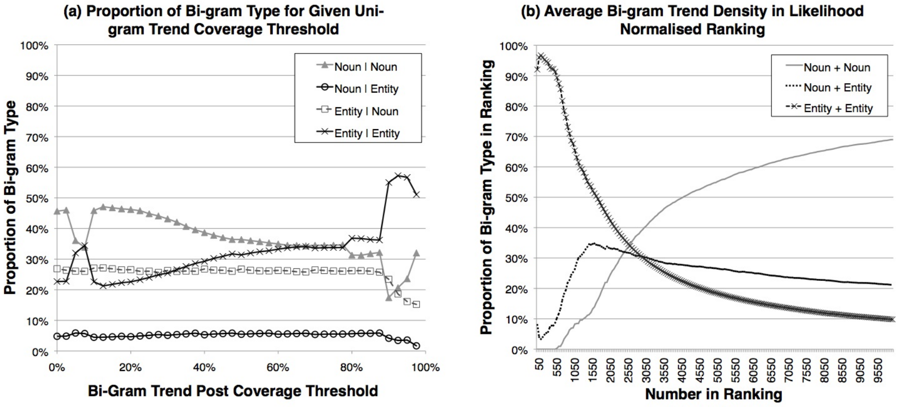


Fig. 3. Bi-gram type densities: (a) as proportion of uni-gram selected blog posts; (b) ranked by ratio of post frequency to expected frequency given independent uni-grams

while the next day news articles reported the playoffs' opening session:

> "Mahan fires 62 to open PGA playoffs"
> - http://sportsnet.ca/golf/2008/08/21/

> "Mahan leads by 4 shots at The Barclays"
> - http://cbs.sportsline.com/golf/story/10942213/rss

Feature specificity would seem to be important to find meaningful and potentially predictive information. Employing methods to refine trending features into the topics that gave rise to them is likely to be beneficial therefore.

5 Feature specificity

One may argue that to impact information about something one must express some relation between it and some other concept. The set of relationships between the key concepts, such as entities, expressed in a document could be said, then, to be the document's topic. Here we employ a similar technique to Alvanaki et al. [8] taking pairs of trending features (a trend bi-gram) to be our key concepts and examining their co-occurrence in source blog postings, the assumption being that frequently co-occurring nouns and named entities would be more likely to be linked topically.

Counts of trend co-occurrences are calculated on a by blog post basis for the day the individual trends occur. Nouns and named entity types are not examined separately here so that the analysis includes finding information across different feature types (e.g. a Person and a Location). If we assume that a topically related collection of documents would contain more than one feature in common and that for trending stories this would include more than one trending feature, we can examine topical consistency in posts selected by a single trending feature and by a trend bi-gram. Correspondingly this will give an indication of how topically specific any single trend feature is.

A bi-gram of each trending feature and its most highly co-occurring trend for each trending feature can be created. The proportion of the uni-gram selected posts the bi-gram appears in gives an indication of how well related the posts are and therefore specificity of the unigram under an assumption of feature independence. The average proportion for bigrams of nouns was found to be 15% (variance 1.6%), but 20% (variance 6.7%) for entity bigrams suggesting entity selected posts are more likely to be topically related. Assuming that the higher the proportion of bi-grams in a sub-set of posts, the more likely that they are to be topically linked, a rank-ordering of proportion of uni-gram selected posts by bi-gram coverage allows bi-gram types to be compared against likely topical coherence. Graph (a) in Figure 3 shows that when bi-grams are present in more than 90% of the posts containing a given uni-gram trend, there is a better than 55% likelihood that both the trending features will be entities, while there is less than 31% likelihood that both trend features in the bi-gram will be nouns.

We estimate specificity by the point-wise mutual information of the bigram features at the document frequency level. This is similar to the topic coherence measure proposed by Newman et al. [26] which sums PMI in term frequency, but calculated intrinsically as in the coherence measure proposed by Mimno et al. [27]. Denoting by $S_t(a)$ the number of posts containing feature a on day t the PMI for a bi-gram feature $\{a, b\}$ is given by:

$$R(a,b) = \frac{S_t(a,b)}{S_t(a)S_t(b)} P_t \tag{3}$$

where P_t is the number of posts with at least one trending feature on day t.

Graph (b) in Figure 3 shows the proportion of each bi-gram type in an averaged rank ordering of the daily bi-gram trends. Bi-grams consisting of two entities are found predominantly at the top of the ranking, followed by those with including one entity. Assuming from the analysis above that bi-grams consisting of entities are more topically specific, this suggests that trending topics can be found by selecting posts containing bi-grams of trending entities scored by how unlikely they are to co-occur at random.

6 Conclusions

This study has considered whether there is evidence to support the idea that social media content could be used to predict or inform news stories. By filtering out trends originating in news, this study has focussed on the small percentage of topics that previous studies, [18], [19], have found *not* to have originated in mainstream news articles. It has also examined whether named entities could be more useful than common nouns as a feature for finding information.

The analysis has shown that named entities, and nouns in general, can be found trending in blog postings that have not previously trended in the mainstream media. On average approximately 12% of these features (18.4% of tagged nouns, 11.1% of tagged entities) subsequently trended in news stories (Although the total number of trending entities exceeded that of nouns). Given 3% of news stories start in social media, this suggest that either the majority of significant social media originated trends are not sufficiently interesting to professional news organisations, or are missed. However, pre-emptive trending features do exist, and simple ranking of trending features by the number of occurrences on the day of trending gives reasonable performance in promoting those that subsequently trend in news. We can conclude, then, that there is potential data in blogs with which to investigate methods for prediction of (or sourcing material for) news stories.

Improved entity tagging, filtering by language could be beneficial. Some trending entities are referred to by different forms of their names which suggests and co-reference resolution could all help in characterising and detecting trends.

Trending entities may not be the subject of topic(s) of interest, and subsequent stories may be indirectly related, on developing events, or take a new angle (a “meta-story” if you will). Examples of all these have been seen in the data examined here. However, not all subsequent trending news stories are related in any significant way to the preceding blog topics sharing the mention, and trends often arise from from multiple topics having the feature in common. These seems to be more likely the case with nouns than with named entities.

Preliminary investigation into topic specificity of trending features has shown that blog posts that are more likely to be related can be identified through co-occurring trending features, and that sets of features are more likely to contain entities than nouns. Selection of bi-grams can be achieved through a rank ordering of how unexpected its co-occurrence is.

Future work will examine whether trend selection or ranking can be improved. It will also look further at trend co-occurrence as it seems like this may be a basis for selecting posts more likely to be topically linked. Clustering of posts with trending features may also be a suitable method. These refinement methods could also assist the observer in assessing the likely interest as a news story. Such topic specificity refinement may also allow sufficient characterisation of trending topics originating in social media to make predictions as to which are likely to be picked up by news media.

References

1. Soboroff, I., Harman, D.: Novelty detection: the trec experience. In: HLT ’05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 105–112. Association for Computational Linguistics, Morristown, NJ, USA (2005)
2. Kleinberg, J.: Bursty and hierarchical structure in streams. Data Min. Knowl. Discov. 7(4), 373–397 (2003)

3. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. In: Proceedings of the 12th international conference on World Wide Web. pp. 568–576. ACM, New York, NY, USA (2003)
4. Gabrilovich, E., Dumais, S., Horvitz, E.: Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In: WWW '04: Proceedings of the 13th international conference on World Wide Web. pp. 482–490. ACM, New York, NY, USA (2004)
5. Franco, L., Kawai, H.: News detection in the blogosphere: Two approaches based on structure and content analysis (2010)
6. Ha-Thuc, V., Srinivasan, P.: Topic models and a revisit of text-related applications. In: PIKM'08: Proceedings of the 2nd PhD workshop on Information and knowledge management. pp. 25–32. ACM, New York, NY, USA (2008)
7. Glance, N.S., Hurst, M., Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics. ACM (2004)
8. Alvanaki, F., Sebastian, M., Ramamritham, K., Weikum, G.: Enblogue: emergent topic detection in web 2.0 streams. In: Proceedings of the 2011 international conference on Management of data. pp. 1271–1274. SIGMOD '11, ACM, New York, NY, USA (2011),
9. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 181–189. Association for Computational Linguistics, Los Angeles, CA, USA (June 2010),
10. Cataldi, M., Di Caro, L., Schifanella, C.: Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining. pp. 4:1–4:10. MDMKDD '10, ACM, New York, NY, USA (2010),
11. Benhardus, J.: Streaming trend detection in twitter. Tech. rep. (2010)
12. Joshi, M., Das, D., Gimpel, K., Smith, N.A.: Movie reviews and revenues: an experiment in text regression. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 293–296. Association for Computational Linguistics, Stroudsburg, PA, USA (2010),
13. Tumasjan, A., Sprenger, T., Sandner, P., Welpe, I.: Predicting elections with twitter: What 140 characters reveal about political sentiment. In: Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (2010),
14. Cha, M., Antonio, J., Pérez, N., Haddadi, H.: Flash floods and ripples: The spread of media content through the blogosphere. In: ICWSM 2009: Proceedings of the 3rd AAAI International Conference on Weblogs and Social Media. AAAI (2009)
15. Simmons, M., Adamic, L., Adar, E.: Memes online: Extracted, subtracted, injected, and recollected. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (2011),

16. Clough, P., Gaizauskas, R., Piao, S.S.L., Wilks, Y.: Meter: Measuring text reuse. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 152–159. Association for Computational Linguistics, Morristown, NJ, USA (2002)
17. Asur, S., Huberman, B.A., Szabó, G., Wang, C.: Trends in social media : Persistence and decay. CoRR abs/1102.1402 (2011)
18. Lloyd, L., Kaulgud, P., Skiena, S.: Newspapers vs. blogs: Who gets the scoop. In: AAAI spring symposium on Computational Approaches to Analyzing Weblogs. pp. 117–124 (2006)
19. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 497–506. ACM, New York, NY, USA (2009)
20. Azzam, S., Humphreys, K., Gaizauskas, R.: Using coreference chains for text summarization. In: CorefApp '99: Proceedings of the Workshop on Coreference and its Applications. pp. 77–84. Association for Computational Linguistics, Morristown, NJ, USA (1999)
21. Thompson, P., Dozier, C.: Name searching and information retrieval. In: In Proceedings of Second Conference on Empirical Methods in Natural Language Processing. pp. 134–140 (1997)
22. Saggion, H., Barker, E., Gaizauskas, R., Foster, J.: Integrating nlp tools to support information access to news archives. In: Proceedings of the 5th Int'l conference. on Recent Advances in Natural Language Processing (2005)
23. Burton, K., Java, A., Soboroff, I.: The ICWSM 2009 Spinn3r Dataset. In: Third Annual Conference on Weblogs and Social Media (ICWSM 2009). AAAI, San Jose, CA (May 2009), http://icwsm.org/2009/data/
24. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology 1. pp. 173–180. Association for Computational Linguistics, Stroudsburg, PA, USA (2003),
25. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370. Association for Computational Linguistics, Stroudsburg, PA, USA (2005),
26. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 100–108. Association for Computational Linguistics (2010)
27. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 262–272. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011),

Information Retrieval and Visualization for Searching Scientific articles and Patents

Lipika Dey, Hemant Gupta, Kunal Ranjan

Innovation Labs
Tata Consultancy Services
Delhi, India
lipika.dey@tcs.com, gupta.hemant@tcs.com

Abstract. Given the rapidly changing face of technology, keeping up with the trends and identifying potential areas to be explored for research or commercialization is a challenging task. Decision makers, research analysts, scholars, research directors all make use of digital collections, use of which is facilitated by search applications developed on top of them. However, search is a human-driven activity and the result of such analysis is largely dependent on the initial inputs that are provided by the expert. Besides, aggregating and assimilating all the information returned by a search engine is no less daunting. In this paper, we propose intelligent methods for presenting search results to help information assimilation. We also present methods for analyzing large collections of documents in an automated way to generate insights that can prove to be useful for analysts. Starting from time-stamped collections of research publications and patent documents, we present several Information retrieval (IR) techniques that can successfully extract and present insights about emerging, popular and receding trends in research along with their current levels of commercialization. We present results of experiments based on research abstracts made available by digital libraries and US patent office.

Keywords: Topic Extraction, Information Retrieval, Commercialization score

1 Introduction

As an off-shoot of the popularity and accessibility of world-wide web there has been a phenomenal rise in the number of research articles, conference proceedings, archived research results, patent filings and grants and several other technical publications which are available online. Though this collection is extremely useful for academic and industrial researchers, searching for any information results in a huge list of documents. Assimilating information from this huge list is a formidable task. While researchers query research collections to understand evolution of an area or topic, state of the art etc. business users may be querying the same collection to remain abreast of the latest in research and look for possible ideas of commercialization to retain competitive edge. In this paper, we present our work towards developing a search and

pp. 265–282; rec. 2015-01-25; acc. 2015-02-25

analytics system that can aid in retrieving relevant information and insight generation from an integrated collection of research publications and patent applications.

Advances in search and information retrieval technology ensure that large volumes of text can be searched efficiently when appropriate queries can be formulated. The focus of text mining research on the other hand has been towards insight generation from large collections of technical documents. Identifying new trends in research topics is a popular area of research. Identifying and exploring relations among research communities is also a popular area. Visualization of information is also a deeply researched area. Some authors have also studied topic evolution over patents and research publications.

However presently there is no search and analytics system that goes beyond listing of articles for an integrated collection of scientific publications and patent documents. Though a listing provides numerical assessment about the potential presence of articles, it does not allow users to easily perceive (a). Content-based relationship among different research areas either at theoretical or application level or (b). the true extent of commercialization of an area or topic. The utility of such a system can be manifold. It can help researchers understand the applicability of research topics. For strategists and decision makers, it would be of help to find yet untapped areas of research and potential areas of new application developments.

The unique aspects of the present paper are as follows:

1. A novel method is presented to identify topic evolution using topically significant phrases, where topics are extracted from time-stamped collections using standard Latent Dirichlet Allocation (LDA). The topical phrases are also used to present a graphical representation of how the underlying topics have evolved or morphed over the years. We have proposed new topic-similarity measures based on Information retrieval (IR) principles that take into account relevance of a document with respect to a topic, rather than word-based measures.
2. The paper proposes new measures to compute the extent of commercialization of a research topic with respect to a patent database. We term this as commercialization score of a research topic. While we have conducted experiments and presented results from the US patent database for the sets of applied and granted US patents over the years 2005 to 2014, the measures are generic and can be used in conjunction to any such database.
3. We present a method for analyzing commercialization scores and commercialization trends to generate insights about further prospects of a topic or an area.

The rest of the paper is organized as follows. Section 2 provides a review of related work. Section 3 discusses how topic similarities are computed to generate a topic evolution graph. Section 4 presents the proposed methods to compute commercialization score and commercialization trends. Section 5 presents some results obtained over a publicly available data set. Finally section 6 concludes with future work.

2 Review of Related Work

A large number of research communities are actively engaged in analyzing scientific articles and patent applications. An interactive prototype system named Action Science Explorer (ASE) was presented in [1], to help researchers with reference management, analyzing topical and citation statistics, text extraction and natural language summarization for single and multiple documents. It supported network visualizations to see citation patterns and identify author clusters. ArnetMiner was proposed in [2]. This paper proposed a unified tagging approach using Conditional Random Fields to generate profile tags for researchers based on publication data extracted from the web. It also proposed a unified topic model called Author-Conference-Topic (ACT) to simultaneously model different types of information in the academic network. Rexplore [4] supports graph-based exploration to understand bibliographic data, research topics and trends. It exploits the Klink algorithm [3] which identifies relations across different research areas using semantically annotated data. [5] proposed several metrics of influence, coverage, and connectivity for scientific literature which can be used to create structured summaries of information, called metro maps. Metro maps are targeted at capturing the developments in a field. An iterative topic evolution learning framework was proposed in [17] based on an inheritance topic model that leveraged citations among documents to analyze topic evolution in an explicit way.

Several groups have also tried to capture researcher communities and group dynamics [6-8] from content and not just from citation. [6] used a key-word based approach to identify topics. In [7] which is an extension of [4], authors proposed the notion of diachronic topic based on communities of people who work on semantically related topics at the same time. It was used to detect events that denote topic shifts within a research community; the appearance and fading of a community; splitting, merging and spawning of new com-munities etc. [8] presents a detailed study on the factors that affect research collaboration among individuals and organizations.

[9] presents a comprehensive literature review on research around analysis of patents. A topic-driven patent analysis and mining system was presented in [10] which studied the evolution of patent network composed of companies, inventors, and technical content using dynamic probabilistic model. It also proposed analytics tools for IP and R&D strategy planning, including a heterogeneous network co-ranking method, a topic-level competitor evolution analysis algorithm, and a method to summarize the search results. [11] proposed an analytical technique called patent trend change mining (PTCM) to capture changes in patent trends. This work, based on association rule-mining was aimed at generating competitive intelligence to help managers develop appropriate business strategies based on their findings. [12] presented a patent analysis system called TechPerceptor which used Natural Language Processing techniques to generate patent maps and patent net-works based on semantic analysis of patents. The system can be used to observe technological hotspots and spot patent vacuums. [13] proposed the use of text mining techniques to develop a Technology Tree(Tech Tree) that can compute similarity scores between patents.

None of the existing systems perform joint analysis of publications and patents using the content of both publications and patent applications. Most importantly none of the systems provide insights about the extent and diversity of research topics and their commercialization to help technology planners.

3 Topic Evolution and Diversification

Latent Dirichlet Allocation [14] is an unsupervised latent variable model that employs Bayesian inference to identify semantic clusters of words in document collections that resemble topics. LDA assumes a range of possible distributions of words with the constraint that they are drawn from Dirichlet distributions. This enables it to learn latent topic models in an un-supervised way ensuring that the topic models are maximally relevant to the underlying data collection. For the proposed work, the LDA model was first applied on yearly collections of publications, which yields topic distribution for each document. Each topic comprises bag of words along with probabilities of each word being generated by that topic.

Word-based representation of topics is useful, but not easy to understand. Instead, the present system adopts phrase-based representation of topic that was proposed in [16]. For each topic, its representative phrases are chosen from among frequently occurring three-grams and two-grams in documents that have a high probability of that topic. Since each document has a probability of each topic being present in it, [16] presented equations to compute the probability of a phrase belonging to a topic based on the occurrence frequency of phrases within documents that contained the topic with a probability greater than a pre-specified threshold. The maximally weighted phrase is used to name the topic. Phrases in the current context refer to N-grams that are faster to compute than natural-language phrases and are also resistant to noise like incorrect grammar or incorrect formatting. N-grams also preserve spatial relationship of words thereby making them closer in appearance to natural language phrases though obtained at much lower computational cost. The frequent n-grams selected to represent a topic are termed as topical phrases. Figure 1 shows phrase based representation of topics that contained the phrase “association rule mining” over the years 2006 to 2009.

Figure 1 shows that a research topic does not remain static over the years. Topics grow, evolve and diversify. A topic’s growth can be tracked by watching the trends in number of publications that continue to cover the topic. Topic evolution can be tracked by watching the changing content. This cannot be tracked using simple word-based representation of topics since the words are difficult to interpret without their context. For example, the word “information” can make many topics look similar, though in reality the topics “Information Retrieval”, “Information Security” and “Management Information Systems” are quite different. Also, new words or phrases emerge and become frequent while old ones phase out. It is therefore proposed that co-occurrences of phrases can better capture continuation and evolution of topics.

Topic diversification captures inter-mixing of topics or adoption of a topic into another topic etc. Figure 2 presents year-wise view of frequently co-occurring N-grams for the query “association rule mining”. It may be noted that the context of “association rule mining” is different from its topical representation shown in figure 2. In fact the 2010 collection did not yield a topic named “association rule mining” though the phrase occurred in the context of “genetic programming” and “traffic prediction”. This obviously indicates that areas like “intrusion detection” or “web traffic prediction” had started adopting association rule mining techniques from 2009 onwards.

We now present a new method to capture topic similarity and then go on to show how this can be used to capture topic evolution and diversity.
Let T_i and T_j represent two different topics of the same year or different years. The topical similarity between T_i and $T_{j,}$ denoted by $\sigma(T_i , T_j)$, is computed in terms of their topical phrases as follows.

Let S_i and S_j be the sets of top n topical phrases associated to T_i and T_j respectively. Let p_i and p_j represent two phrases where $p_i \in S_i \; and \; p_j \in S_j$.

Let $D_i \; and \; D_j$ denote the collections of documents that contain p_i and p_j respectively. $D_i \; and \; D_j$ may be identical, overlapping or completely disjoint. The degree of overlap of these two sets captures the *neighborhood similarity* of p_i and p_j, denoted by $\eta(p_i, p_j)$ and is computed as follows:

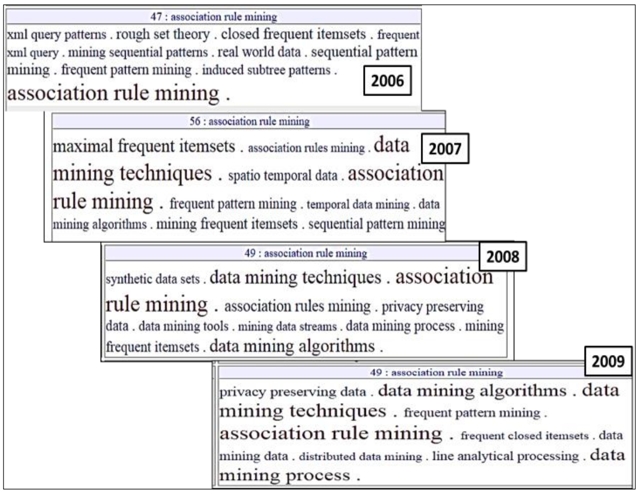
$$\eta(p_i, p_j) = \frac{|D_i \cap D_j|}{|D_i \cup D_j|} \tag{2}$$

For each phrase $p_i \in S_{i,}$ let $\alpha_j \epsilon S_j$ be the phrase with maximum value for $\eta(p_i, \alpha_j)$ i.e. $\eta(p_i, \alpha_j) \geq \eta(p_i, p_j) \; \forall p_j \in S_j$. In other words, the phrase p_i of topic T_i co-occurs maximally with α_j of T_j. Similarly, for each phrase $p_j \in S_j$ let $\beta_i \epsilon S_i$ be the phrase with maximum value for $\eta(\beta_i, p_j)$ i.e. $\eta(\beta_i, p_j) \geq \eta(\beta_i, p_j) \; \forall \beta_i \in S_i$.

It is obvious that the neighborhood similarities for a pair of phrases are not symmetric in nature. The similarity between a pair of topics is computed as the average neighborhood similarity between all pairs of topical phrases for pair.

$$\sigma(T_i , T_j) = \frac{1}{2n}\left(\sum_{i=1}^{n} \eta(p_i, \alpha_j) + \sum_{j=1}^{n} \eta(\beta_i, p_j)\right) \tag{3}$$

It may be noted that unlike most similarity measures that are computed on the basis of shared words or terms, $\sigma(T_i , T_j)$ computes similarity of topics in terms of shared documents in which representative terms of T_i and T_j co-occur.



2006

47 : association rule mining

xml query patterns . rough set theory . closed frequent itemsets . frequent xml query . mining sequential patterns . real world data . sequential pattern mining . frequent pattern mining . induced subtree patterns . association rule mining .

2007

56 : association rule mining

maximal frequent itemsets . association rules mining . data mining techniques . spatio temporal data . association rule mining . frequent pattern mining . temporal data mining . data mining algorithms . mining frequent itemsets . sequential pattern mining

2008

49 : association rule mining

synthetic data sets . data mining techniques . association rule mining . association rules mining . privacy preserving data . data mining tools . mining data streams . data mining process . mining frequent itemsets . data mining algorithms .

2009

49 : association rule mining

privacy preserving data . data mining algorithms . data mining techniques . frequent pattern mining . association rule mining . frequent closed itemsets . data mining data . distributed data mining . line analytical processing . data mining process .

Fig. 1. Topic - "Association Rule Mining" phrases through years 2006 – 2009

Topical evolution is captured through intra-year and inter-year topic similarity matrices. An intra-year topic similarity matrix captures pair-wise topic similarities for topics belonging to the same year. An inter-year similarity matrix captures pair-wise similarity for topics of consecutive years. Thus for a time-stamped collection containing articles published over N consecutive years, we obtain N intra-year similarity matrices and N-1 inter-year similarity matrices, each of k^2 dimension, where k is the number of topics per year.

The similarity-matrices constructed as above can be considered as adjacency-matrix representation for a multi-layered labeled and weighted graph G in which nodes represent topics. Each layer contains nodes representing topics of the same year. Nodes within a single layer are connected by weighted, undirected edges where the weight of an edge is equal to the similarity of the topics connected by it. Absence of an edge indicates no similarity. A pair of nodes from two different layers is connected by a weighted edge if the layers denote consecutive years. The weight is again equal to the similarity of the two topics it connects. A node in this graph is denoted by T_i^m where i is a topic index and m is a year-index.

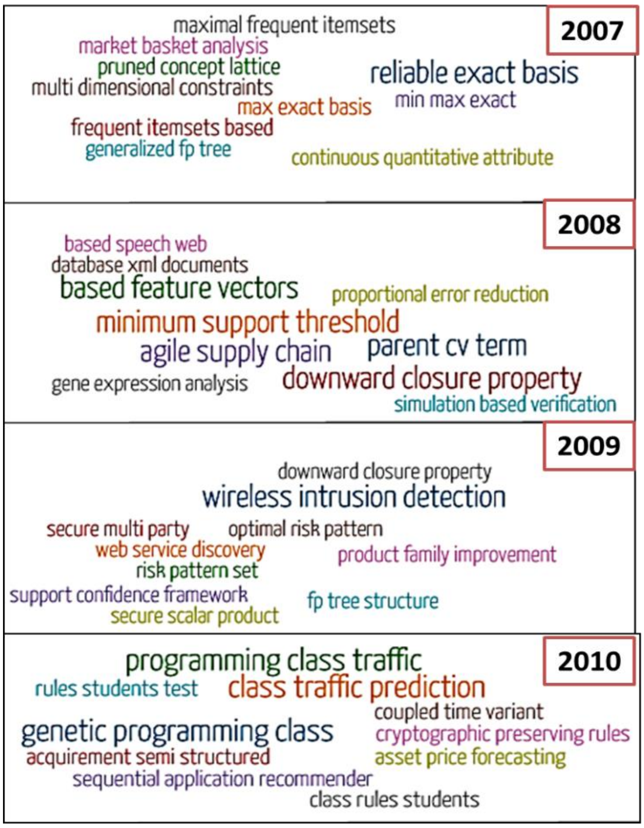


Fig. 2. Evolution of "Association Rule Mining" through contextual collection of topical phrases

It is proposed that topic evolution and diversification can be obtained as strongly connected components of the above graph. The algorithm proposed below finds strongly connected components within the layered graph. It uses two parameters ε and κ, which are defined below.

Definition 1: ε is defined as the *similarity_threshold* of the topic evolution graph. Two topics T_i, T_j are considered to be ε-related if and only if $\sigma(T_i, T_j) >$ ε. ε-related does not imply that one topic has evolved from another topic. It is the minimum requirement for evolution.

Definition 2: κ is defined as the *connectivity_threshold* for topic evolution. The value of κ lies between 0 and 1. A set of λ nodes are said to be κ_connected to each other, provided each of them is ε-related to at least κλ number of nodes from this set. When κ is equal to 1, the set of nodes are fully connected to each other.

We now explain the algorithm to find κ_connected components of ε-related topic-similarity matrix.

4. Input ε and κ. Initialize C to NULL. C will finally contain a set of independent components, where each component will denote a set of connected topics.

5. Let B = (D, E) be a sub-graph of G which is constructed as follows. E contains only those edges of G which satisfy the following condition

$$\sigma(T_i^m, T_j^n) > \ \varepsilon\, AND\, \left((\,n = m)\; OR\; (n = m + 1)\right)$$

Consequently, D contains only those nodes of G, which have at least one ε-related edge incident on it. In other words B contains all topic nodes that are ε-related to at least one more topic within the same year or across consecutive years.

6. For each edge in B, the weight $\sigma\left(T_i\,, T_j\right)$ is now recomputed as follows:

$$\sigma\left(T_i\,, T_j\right) \ = \sigma\left(T_i\,, T_j\right) /\nu \qquad (4)$$

where ν is the maximum of degrees of T_i and T_j. This reduces the weight of those edges that are connected to nodes which in turn are ε-related to many other nodes. Topics that represent generic and basic areas may overlap with many areas. Edges emanating from these topics get less priority. This step helps in suppressing noisy and obvious evolutions while giving priority to area-specific evolutions.

7. Arrange edges of B in decreasing order of associated weight $\sigma\left(T_i\,, T_j\right)$.
8. Remove the first element of B and initialize a cluster C with this element.
9. Repeat steps a to d until B is empty
   (a) Remove the top-most element ***e*** of B.
   (b) Add ***e*** to an existing cluster X of C if its addition maintains the κ connectivity in C. If ***e*** satisfies this relation with more than one cluster of C, add ***e*** to all such clusters.
   (c) Otherwise start a new cluster C'.
   (d) Update clusters $C = C\ \cup C'$
10. Output C.

The output of the above algorithm is a graph of connected components, where each component is a layered graph. A visualization of the graph is generated in which each layer is assigned a unique color. The layers are then presented in terms of increasing index of years from left to right.

Figure 3 illustrates two independent clusters from the topic evolution graph that was generated using all topics extracted from publications from 2007 to 2012. The bigger cluster shows the relationship of the areas Natural Language Processing (NLP), semantic web, gaming systems, online learning systems and social networks. This is obviously a correct and interesting evolution. It illustrates the continuing and important applications of natural language processing techniques to game-based learning and intelligent tutoring systems. The second cluster in figure 4, lower right corner shows continuing interest in support vector machines as a stand-alone topic.

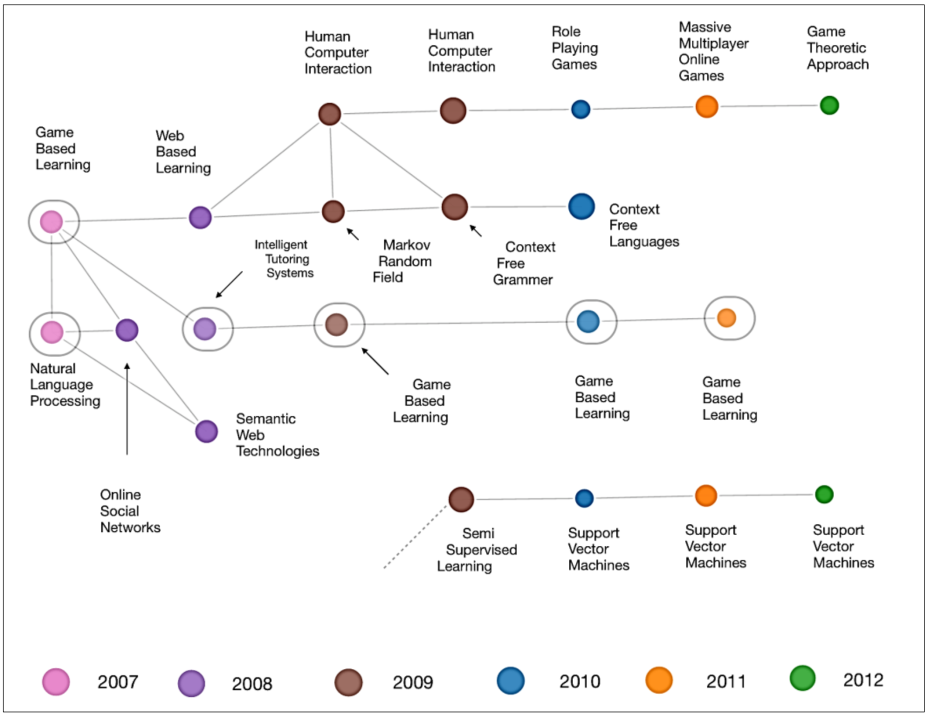


Fig. 3. Topic Evolution Graph (partial)

Table 1 summarizes the evolution history for a few popular topical phrases, using our method and c-ITM proposed in [17]. Most of these phrases were found by [17]. Columns 2 and 3 show the top phrases in predecessor and related topics of same or later years identified by the proposed method. Column 4 shows the topic names given for related topics as presented in [17]. Column 5 shows manual judgment about the relationship between these human-assigned topic names given by c-ITM and the topic phrases yielded by the proposed method.

Table 1: Topic Evolution - comparing proposed method with c-ITM

Top topic phrases and the year	Topical phrases of Predecessor Topics (proposed method)	Topical Phrases of Related topics - Contemporary or Later (proposed method)	Predecessor Topic as per c-ITM [17]	Our observation
ad hoc networks (2000)	Intersymbol interference isi(1995), Ahn collision detection(2000), Multicast rout-	Vehicular ad hoc(2006), Heuristics analytics system(2007)	Network communication since 1994	All phrases in columns 2 and 3 related to Network Communication

	ing protocols(2000)			
wireless sensor (2002)	Wavelength division multiplexing(2001) , Bluetooth 1.1 (2001), Sensor network systems(2002)	Underwater Sensor networks(2006) , Ubiquitous computing technologies(2007) , IP multimedia subsystem(2007)	sensor networks since 2003	Sensor Networks as a phrase was detected in 2002. Lot of related phrases were detected in 2001.
content based image retrieval (1995)	Markov random fields(1995) , Optical flow fields(1995)	Shear warp algorithm(2002) , Remote sensing image(2007), Context intelligent diagnosis(2009) , Medical image segmentation(2011)	hidden in information retrieval from 1993	Topical phrases in column 2 show evolution from Image Processing, Graphics and Hidden Information Retrieval
intrusion detection (2002)	Virtual private networks(2001), Denial service attacks(2001)	Access control policies(2003) ,	protocol security since 2000	Topical phrases are related to Protocol Security
support vector(2001)	Neural networks(2000) , Self organizing map(2000) , Principal component analysis(2000)	Hidden markov models(2002), Facial expression recognition(2006)	neural network since 2000	Topical phrases show evolution from Neural Networks
semantic web (2004)	Xml powered web(2003) , Web usage mining(2003)	Intelligent tutoring system(2005) , Service oriented computing(2006) , Web ontology language(2007), Formal concept analysis(2007) , Social text Streams(2007)	evolved from knowledge ontology since 2002	Topical phrases of predecessor and related topics depict significance of web ontology and xml based web architecture to semantic web
signature scheme (1995)	Public key infrastructure(1995) , Role based Access(1995) , Access control mechanisms(1995)	Buffer overflow attacks(2003) , Stolen verifier attack(2003), Key management system(2006)	protocol security since 2004	Significant methods/technologies related to protocol security emerge through Topical phrases of predecessor and related topics
fading channels(2000)	Code-division multiple access(1995), Bit error rate(1995),	Multiple access interference(2002)	channel coding since 2004	Topical phrases show evolution from channel coding

	Channel Impulse response(2000)			
xml data (2000)	Synchronized multimedia integration(2000)	Jsp xml web(2002) , Database management systems(2005) , Nearest neighbour queries(2005)	evolved from database since 2003	Role of database in the emergence of xml data formats is visible
energy consumption(2007)	Wireless sensor networks(2007), Sensor network applications(2007)	Dynamic voltage scaling(2007) , Pervasive computing environments(2008) , Energy harvesting systems(2010)	N/A	

4 Computing Commercialization Score of Topics

We now present a method to compute and present to the end-user a comprehensive view about the current state of commercialization of a research topic based on the patent volumes and patent trends applied in the area. Each topic is assigned an aggregate commercialization score based on its strength in an associated collection of patent applications. Presently, we have considered all patent applications that have been filed and/or granted with USPTO during the period of 2005 to 2013. However, the proposed method is generic and applicable for any collection.

Patents are also time-stamped documents. Each patent document is first subjected to phrase extraction. All 2-grams and 3-grams are extracted and used for indexing the patents. The Lucene indexer is used for the purpose of indexing and retrieving patent documents for a given topic.

Let T_i be a research topic belonging to the year y generated from publication analysis. Let S_i be the set of n topical phrases representing T_i. Let $\Psi(T_i)$ represent the commercialization score of T_i which is computed using an aggregated relevance score of the documents that are retrieved by Lucene for phrases in S_i as follows.

Let P_i denote all patent documents that contain at least one phrase from S_i. A document is said to contain a phrase if all the words of the phrase are found to lie within a window of w words in the document.

For each document retrieved by Lucene $d_i \in P_i$ relevance of d_i to topic T_i, $R(d_i, T_i)$ is computed as follows

$$R(d_i, T_i) = pFactor(T_i, d_i) * tNorm(T_i) * \sum_{p \in S_i}(f(p, d_i)^2 * I(p)^2 * rank(p))$$

where $f(p, d_i)$ term frequency of p in d_i and $I(p)$ is the inverse-document frequency,

$rank(p)$ is normalized significance of phrase p where the most significant phrase in S_i has maximum significance, pFactor(T_i, d_i) is the normalized score based on how many phrases of S_i are found in d_i, where the document that contains most topical phrases receives maximum weights,

tNorm(T_i) is a normalizing score computed as follows:

$$tNorm(T_i) = \frac{1}{\sqrt{sumOfSquaredWeights}}, where\ sumOfSquaredWeights = \sum_{p \in S_i} I(p) \quad (5)$$

A document may contain phrases belonging to more than one topic, though its relevance to each topic may differ. tNorm ensures that the document is considered more relevant to a topic T_x (say) than another topic T_y (say), if it contains highly significant phrases of T_x but less significant phrases of T_y.

Finally $\Psi(T_i)$ is computed as the logarithmic transform of the aggregated relevance scores of all documents containing topical phrases of T_i as

$$\Psi(T_i) = \log(\textstyle\sum_{d_i \in P_i} R(d_i, T_i)). \quad (6)$$

The commercialization score is further discretized into a 5 point scale, using equal discretization over all non-zero scores, and are denoted by VERY HIGH, HIGH, MEDIUM, LOW and VERY LOW. Figure 4 presents a heat-map that illustrates the extent of commercialization for each of 100 research topics of 2012. The text pop-up shows that the topic of wireless-sensor networks has been heavily commercialized.

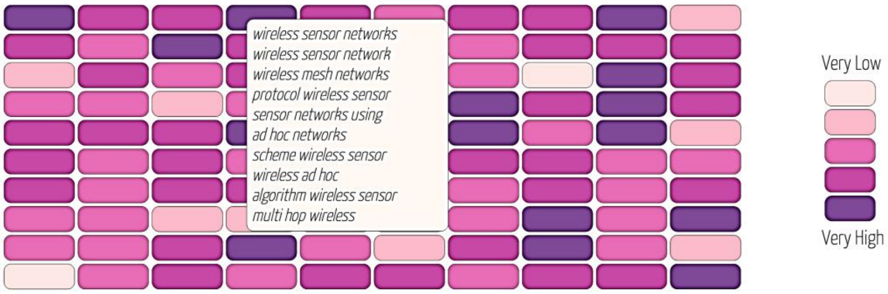


Fig. 4. Heat map showing aggregated commercialization of research topics of 2012. Wireless sensor Networks have been heavily commercialized

Figure 5 presents a graph, each of whose nodes are topics that represent the area of Wireless Sensor Networks, which is the same as the topic-evolution graph component for the area, with one difference. The size of a node in this graph is proportional to its commercialization score. The number of nodes in a particular year is indicative of the diversity of the topic as a whole. This graph also depicts that interest to file patents in this area had reached its peak in 2010.

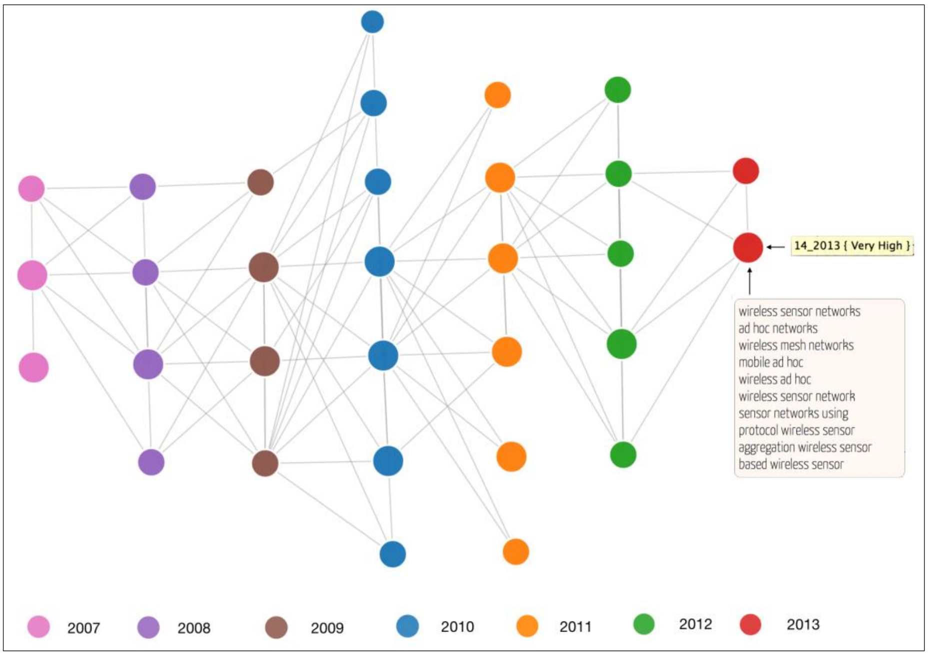


Fig. 5. Commercialization **trends of topics related to "Wireless Sensor Networks"**.

4.1 Analyzing Commercialization Trends

The history of commercialization of research topics can further lead to an understanding and categorization of commercialization of research areas into emerging, receding or yet-to-be-explored for potential commercialization. In order to detect trends, for a particular topic, say T_i^m, for a given year *m*, we first find all topics of past years that are maximally related to T_i^m using the topic evolution graph. Let L_i^m denote this list. Year-wise commercialization score for T_i^m is then computed using aggregate commercialization scores for all topics in L_i^m with the document collection restricted to those patent applications that have been filed in the year *m* only. Thus the yearly commercialization score for a topic T_i^m is given by

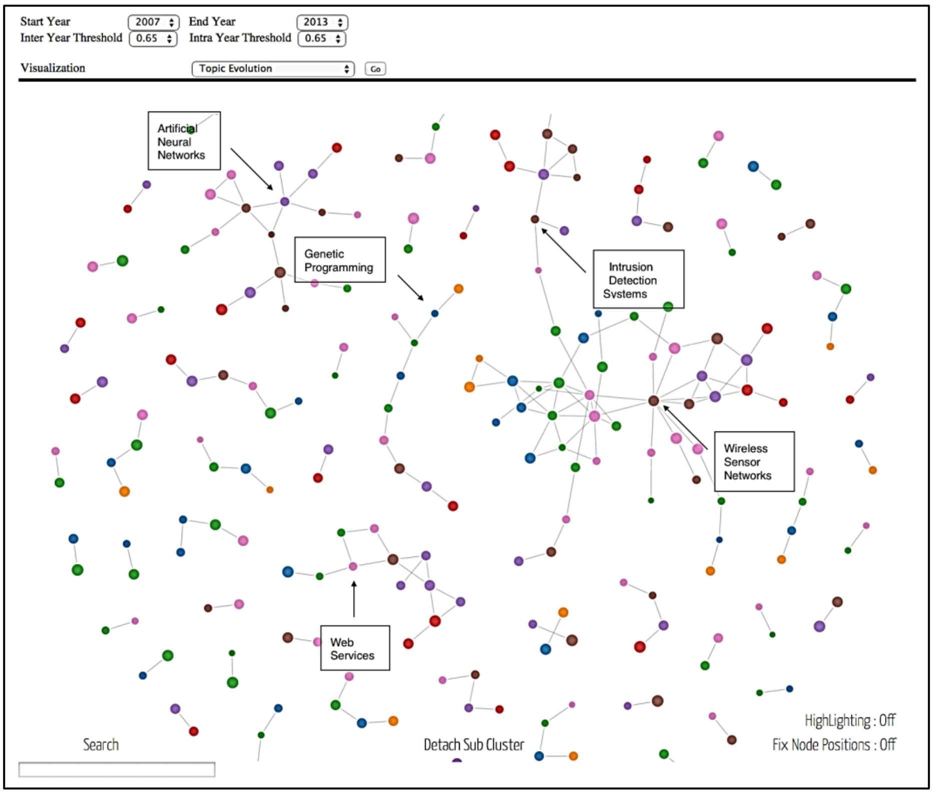
$$C(T_i^m) = \sum_{d_i^m \in P_i^m,\ t \in L_i^m} R(d_i^m, t) \tag{7}$$

where P_i^m denotes the collection of patent applications that have been filed in the year *m* and contains at least one topical phrase from the topics in L_i^m.

The total commercialization score along with trends of yearly commercialization scores are used for insight generation.

5 Experiments and Results

In this section we present some results from an implementation of the proposed methods to design a search system. The system has been implemented over a SOLR[1] based platform as a web-service. Research abstracts for the purpose were collected from sites dl.acm.org and csxstatic.ist.psu.edu/about/data, which have been made available by ACM and Citeseer respectively. The collection contains abstracts of Computer Science related publications along with title of paper, authors, venue and date of publication. After crawling, cleaning and indexing, the data has been stored locally on a server. All the proposed analytical methods run off-line to generate the similarity matrices and commercialization scores. Users can access the system as a web-application to search, drill-down and also see visualizations of topic evolution, commercialization etc. through appropriate inter-active visualizations.



Start Year 2007 End Year 2013
Inter Year Threshold 0.65 Intra Year Threshold 0.65
Visualization Topic Evolution

HighLighting : Off
Fix Node Positions : Off

Fig. 6. Topic Evolution Graph (Partial)

[1] Lucene.apache.org/solr

Figure 6 presents a few components of the topic evolution graph generated from all the topics for all years. Few nodes from two components have been highlighted. The first component on the left shows how areas of machine-learning and biological data mining have interacted over the areas. Similarly the component on the right shows that wireless technologies and privacy and security related research have influenced each other.

Table 2 presents the most commercialized research topics yielded by the system using the proposed commercialization scores, where research topics are extracted from the research publications. Table 3 presents some actionable insights generated from analysis of commercialization trends as stated at the end of section 4. On extreme left the column shows research areas that are hot, have commercial potential and not yet fully exploited. The second column indicates areas which are well-established and commercialization is on the rise. The third column shows areas that are very well-explored and saturated with patents and thus may be highly competitive to enter at this point. The fourth column shows areas which are theoretically well-explored and show declining trend of patenting.

Table 2. Top 10 most commercialized research topics (2005 - 2013)

Top 10 most Commercialized topics (2007 - 2013)
Using Mobile Devices
Cryptography
Wireless Sensor Networks
Real Time Systems
Image retrieval
Brain Computer Interface
Predictive Control for Autonomous Vehicle
Embedded Systems
Reduced Power Consumption
Intrusion Detection System

Table 3. Insights generated from Analysis of commercial Trends

New Research Areas - Very Few Patents - Rising Patent Trend	**Hot Research Areas - Many Patents - Rising Patent Trend**	**Popular research area – Large number of patents - Steady Patent Trend**	**Receding Research Area - Many patents - Patent Trend Decreasing**
Wheeled Mobile Robot	Multi-agent Systems	Wireless Sensor Networks	Collaborative Filtering
Human Robot Interaction	Support Vector Machines	Semantic Web	Service Oriented Architecture
	Social Network Analysis	Time Series Classification	
	Artificial Neural Network	Error Correcting Codes	

	Magnetic Resonance Imaging	Information Security	
	Cyber Physical Systems	Commercial Cloud Services	
	Electronic Health records		

Figure 7 presents the list of top 20 companies in USPTO database which have filed maximum patents in the areas listed in Table 2 between 2005 and 2013 along with the number of patents filed by them. Figure 8 (left) presents the most frequently occurring 3-gram phrases in patent applications for top 3 companies. On the right it presents phrases from patents by 3 companies which have filed a large number of patents in the areas listed in Table 1 only, though do not appear in the list of Figure 7. This shows an interesting aspect of commercialization. These are niche companies filing patents in specific trending areas of research. The established companies have a more diverse portfolio which includes many well-explored areas of research.

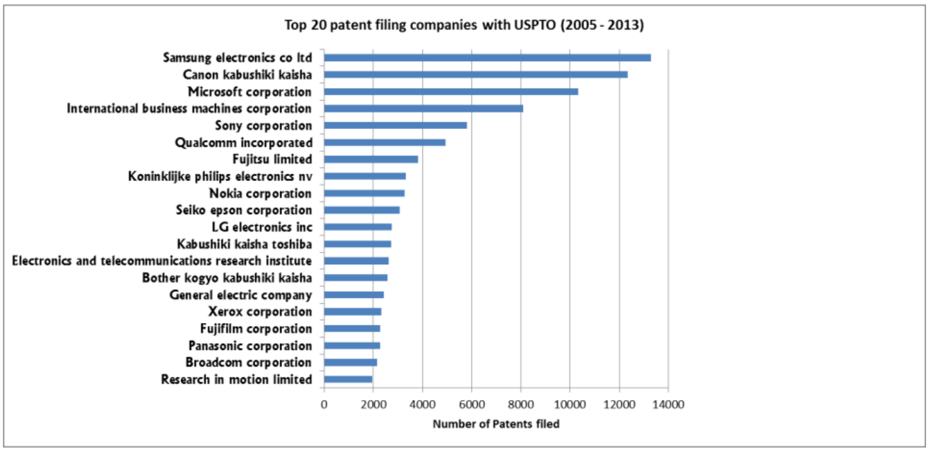


Fig. 7. Top 10 Companies filing patents in the above areas

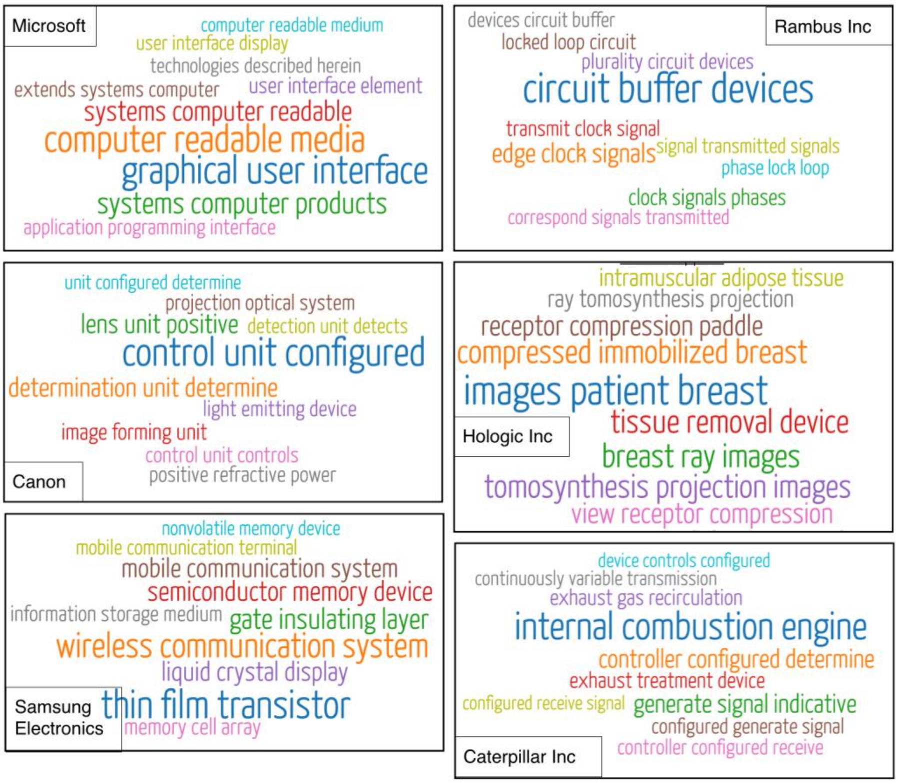


Fig. 8: Patent profile of companies through frequent phrases

6 Conclusions

In this paper, we have presented methodologies for analysing large volumes of research publications for information gathering and insight generation. We have presented results from an instance of implementation which currently analyses hundreds of thousands of research abstracts and patent applications jointly. The objective of the joint analysis is to come up with insights about current states of commercialization of research areas. Such a system helps in understanding current state of research as well as look for new ideas of commercialization. It also helps in understanding the existing competition.

Our future work lies in complete automation of the decision making process by aligning the content with external hierarchical indexing mechanisms like Wikipedia, journal content hierarchy etc. to explore inter-disciplinary topical relationships. This will help in better understanding of application of research areas and technologies to different areas for better decision making purposes.

References

1. Dunne, C., Shneiderman, B., Gove, R., Klavans, J., & Dorr, B. (2012). Rapid understanding of scientific paper col-lections: Integrating statistics, text ana-lytics, and visualization. Journal of the American Society for Information Sci-ence and Technology, 63(12), 2351-2369.
2. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data min-ing (pp. 990-998).
3. Osborne, F., & Motta, E. (2012). Min-ing semantic relations between research areas. In The Semantic Web–ISWC 2012 (pp. 410-426). Springer Berlin Heidelberg.
4. Motta, E., & Osborne, F. (2012). Making Sense of Research with Rexplore. In 11th International Semantic Web Conference ISWC 2012 (p. 49).
5. Shahaf, D., Guestrin, C., & Horvitz, E. (2012, August). Metro maps of science. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data min-ing (pp. 1122-1130).
6. Yan, E., Ding, Y., Milojević, S., Sugimoto, C.R.: Topics in dynamic re-search communities: An exploratory study for the field of information retrieval. Journal of Informetrics, 6(1), 140-153. (2012)
7. Osborne, F., Scavo, G., & Motta, E. (2014). Identifying diachronic topic-based research communities by clustering shared research trajectories. In The Semantic Web: Trends and Challenges (pp. 114-129). Springer International Publishing.
8. Bozeman, B., Fay, D., & Slade, C. P. (2013). Research collaboration in universities and academic entrepreneurship: the-state-of-the-art. The Journal of Technology Transfer, 38(1), 1-67.
9. Abbas, A., Zhang, L., & Khan, S. U. (2014). A literature review on the state-of-the-art in patent analysis. World Pa-tent Information, 37, 3-13.
10. Tang, J., Wang, B., Yang, Y., Hu, P., Zhao, Y., Yan and others (2012). Patentminer: topic-driven patent analysis and mining. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1366-1374.
11. Shih, M. J., Liu, D. R., & Hsu, M. L. (2010). Discovering competitive intelligence by mining changes in patent trends. Expert Systems with Applications, 37(4), 2882-2890.
12. Park, H., Kim, K., Choi, S., & Yoon, J. (2013). A patent intelligence system for strategic technology planning. Expert Systems with Applications, 40(7), 2373-2390.
13. Choi, S., Park, H., Kang, D., Lee, J. Y., & Kim, K. (2012). An SAO-based text mining approach to building a technology tree for technology planning. Expert Systems with Applications, 39(13), 11443-11455.
14. David M. Blei, Andrew Y. Ng, and Mi-chael I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Re-search, 3:993–1022, 2003.
15. T. L. Griffiths, M. Steyvers, D. Blei, and J. B. Tenenbaum. Integrating topics and syntax. Advances in Neural In-formation Processing Systems (2005).
16. Dey, L., Mahajan, D., & Gupta, H. (2014). Obtaining Technology Insights from Large and Heterogeneous Document Collections. In Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on (Vol. 1, pp. 102-109).
17. Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and C. L. Giles. Detecting Topic Evolution in Scientific Literature: How Can Citations Help? In CIKM, 2009.

Summarization of Technical Articles: Modeling User's Expectation from a Summary Using Specificity Score

Shailesh Deshpande and Athiappan G.

Tata Research Development and Design Centre
{shailesh.deshpande, athiappan.g}@tcs.com

Abstract. Conventional importance based extractive summarization methods face many difficulties as notion of importance is malleable. Instead, users expectation from a summary could possibly be defined in more precise manner – say by some of the discourse properties. In this paper we use specificity score – a measure of how specific or generic a particular text is – to characterize the types of documents, and further encode the expectation from a summary. We further demonstrate use of specificity score to summarize technical articles. Our hypothesis is: users expect summary sentences to convey more specific information from a technical article.

1 Introduction

Extractive summarization techniques create summaries by selecting sentences that are important - in some sense - to the document. In abstractive summarization also, important sentences are extracted and then subsequently paraphrased to the required length of summary. Many techniques assign the importance to the sentences and order them accordingly to select top scoring *k* sentences as a summary. There are many methods for deciding the importance of the sentences: simple word frequency based, key words based, centrality based and so on.

Conventional - importance based summarization - has two difficulties: First, deciding important information for summary is a nontrivial task. Notion of importance is malleable and subjected to change based on point of view. Disagreement between summaries by experts is a well-studied behavior [**1**], [**2**], [**3**], [**4**]. Disagreement between experts on potential best summary affects the summary evaluation task as well. So instead of taking one model summary for evaluation, summaries from two three experts (may be more) are taken for comparison with peer summary [**5**], [**6**], [**7**]. Second, the sentences extracted might be important but not the expected one. The expectation from the summary sentences can be expressed in the form of discourse relations (or any other suitable property). For example; contradictory sentences should not be extracted, sentences should provide specification and so on. Traditional summarization techniques do not incorporate the mentioned expectation measure in the process. The mismatch between summaries by experts and summaries by importance based algorithms [**8**] is because of expectation mismatch.

pp. 283–294; rec. 2015-01-28; acc. 2015-02-27

A case in this point: Technical articles. Generally speaking, we expect more specific sentences from a summary of technical articles such as scientific publications, research report *etc.* Table 1 shows summary of Relative Utility (RU) paper by Radev *et. al.* [**2**] generated by one of the standard algorithms. These selected sentences are from the list of 15 (~15 %) sentences extracted as a summary by Lexrank algorithm [**9**]. They are ordered according to the sequence number of the sentence in the original document.

Table 1. Lexrank summary of RU paper

The main problem with traditional co-selection metrics (thus named because they measure the degree of overlap between the list of sentences selected by a judge and an automatically produced extract) such as Precision, Recall, and Percent Agreement for evaluating extractive summarizers is that human judges often disagree about which the top n% most important sentences in a document or cluster are and yet, there appears to be an implicit importance value for all sentences which is judge-independent.

We have measured the utility correlation for three judges on 3,932 sentences from 200 documents from the HK News corpus.

We will call this observation the principle of Summary Sentence Substitutability (SSS).

From the extracted sentences we can see that the sentences are important and introductory (introducing topic to the reader) or generic. If reader is interested in knowing most prevailing topic, or generic discussion in the paper, then this summary would perform reasonably well. But on the other hand, if reader expects more detailed information such as research findings, comparative assessment of method and so on then the sentences do not convey required information. Reader would expect sentences such as these (Table 2):

Table 2. Alternative expected summary sentences for RU paper

The average value of R across all documents at the 5% target length is 0.598 while the average value of J is 0. 799. The corresponding values for the 20% target length are R = 0 635 and J = 0.835.

Second, MEAD and WEBSUM score approximately the same on all metrics with MEAD doing slightly better on the Word overlap, Bigram overlap, and longest common subsequence measures and WEBSUM on the cosine metric.

In this paper we propose to use specificity of the sentences as a criterion to select the sentences for summary. We take two important types of documents; news articles and technical articles (papers) and show how specificity score can be used for summarization and for modeling expectation from a summary by the reader. The features for specificity (of a sentence) used in this paper are very similar to work by Deshpande *et. al.* [**10**], [**11**] and the work by Louis *et. al.* [**12**].

What is generic (information) and what is specific is less subjective than importance. The disagreement between experts on classifying sentences into specific and generic class is quite low [**11**] compared to the disagreement on important sentence (selected for summary). Moreover, summary can be determined by nature of information imparted by the sentences that is if generic, or more specific, or a healthy mix of generic and specific sentences is required.

Predominantly, research in summarization is driven by intrinsic properties (for example, importance of a sentence) of the document. Whereas, using extrinsic criteria for driving summary process completely might be still impractical, deciding how specific or how generic information the summary should have - for performing a particular task - is a viable option. Only specificity score might not be sufficient to convey all the aspects of extrinsic measure, but it certainly helps in expressing the expectation from summary in more objective way – which in turn can be used to drive the summarization process.

2 Related work

The feature model for specificity for document understanding tasks first appear in [**10**], [**11**], and then in [**8**], [**12**]. Both of these models use very similar features such as length, semantic depth, named entities (NE) to characterize the specificity of a sentence. Louis and Nenkova [**12**] use supervised learning approach to classify sentences from model and peer summaries into specific and generic. Whereas, Deshpande *et. al.* [**10**], [**11**] use unsupervised approach to rank the sentences according to their specificity and then select top k sentences as very specific feedback from customer comments. Though these studies indicate complete model for specificity of the sentences and direct usage of specificity score for some of the document understanding tasks, earlier researchers indicate – explicitly or otherwise - need for studying nature of information (in sentences) for its generic and specific tendencies.

Jones [**13**] in her important work on the term indexing for information retrieval task, argues that indexing term specificity should not be decided semantically but rather should be defined statistically. Thus, highlights that the words appearing less frequently tend to be specific.

Hassel [**14**] tests his hypothesis, that sentences containing NE would be more important for summary without much success. The summaries created with NE feature do not show improvement in recall. The lack of improvement in recall rather decrease in recall can be explained using specificity scores: NE indicates more specific sentences (in general) than generic sentences. Naturally, evaluation of such summaries with model summaries with more generic sentences is bound to perform poorly - as in this case.

Halteran *et al.* [**1**] propose use of factoids – self-contained information units – for summary evaluation and study extensively how factoids from different reference summaries can be used to create consensus summary. In the study, they found more general factoids in reference summaries than more specific factoids (similar observation is reported by Louis-Nenkova [**8**]). The study reveals two observations that are important in present context: First, human tendency towards expressing facts in the documents (news articles in this case) in more general way than specific and second, intuitive hypothesis that importance and generalization are inversely proportional to each other (for some type of documents).

Jing et. al. [**15**] use three step process in producing summary – sentence extraction, sentence reduction, and sentences recombination. During the sentence recombination step, one of the substitution operations they suggest is to replace sentence (or its part) with more general or more specific information. They identify rules for these substitutions by manually analyzing human summaries.

Further discussion is organized as follows: Section 3 provide higher level approach for summarization studies using specificity score. Section 4 discusses summary generated by algorithm and its analysis. Section 5 concludes the work.

3 Methodology

3.1 Objectives

Broad level goal for the specificity experiments for summarization is to find the mapping between summarization factors and specificity – that is given a summarization factor (say PURPOSE – audience and use) [**16**], can we define specification of summarization system in terms of specificity? Specific goal for present work is to study a) how specificity score vary for summaries of different types of documents, b) how specificity vary for news articles with single lead (single story and its details) and multiple leads (single story with multiple sub-stories with its own details)? c) What are the characteristics of lead sentences (which are good summary sentences) in terms of specificity? We want test our hypothesis: a) more specific sentences are expected as a summary for some types of news reports (such as finance, interesting court or electoral cases, natural calamities), b) generic sentences are expected for research paper summary which is very close to the abstract of a paper, c) specific sentences form good summary for information such as research findings, scientific claims and so on.

3.2 Algorithm

In this section we provide glimpses of the specificity features and algorithm (Table 3) and describe how the specificity score is used for document understanding and summarization tasks. We show the results on news articles and technical papers. First, we calculate specificity scores for sentences of article of our interests. The plot (Fig. 1) of specificity score and sentences number reveals structure of the document in terms of specificity of information provided by each section. These plots are used for analyzing intentional structure of the document.

We begin with extracting various semantic and statistical features of a word and then of a sentence. Semantic Depth (SD) measures number of edges, in the hypernym tree from the WordNet [**17**], between the root word and a given word. For example, apple is more specific word than the word “fruit”. Average Semantic Depth (ASD) is a sentence level metric that measures average semantic depth of all the words in that sentence. Semantic Height (SH) is reciprocal of Semantic Depth and it measures number of edges from the leaf node in the hypernym tree from WordNet [**17**]. SH is averaged over all the words in a given sentences to give Average Semantic Height (ASH). Total Occurrence Count (TOC) measures how many times a word occurs in the ontology such as WordNet [**17**]. More specific words tend to occur less frequently (generally speaking with a few exceptions) in the WordNet like ontology. We take three lowest count words and sum TOC of them to indicate TOC for a given sentence. Named Entity (NE) count and length are simple measures indicating number of NEs, and length (number of words) for a given sentence respectively (Please see [**11**] for detailed implementation). After calculating specificity scores for each sentence, summarization can be approached as follows:

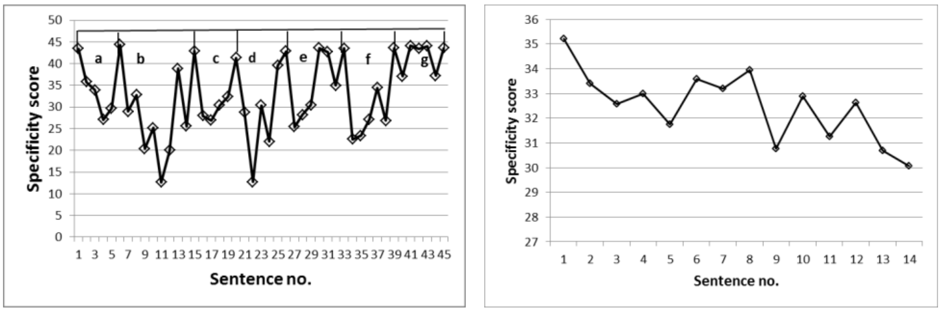
1. Specify the expectation from a summary using specificity score, that is, if user expects more generic information or specific information. For technical articles if abstract like summary is required extract generic sentences. If the technical summary is required, extract more specific sentences.
2. Sort the sentences according to the expectation set by user: For technical summary sort sentences by descending order of the specificity score (larger values first).
3. Choose top *k* sentences as per the length or set the threshold for specificity as per the requirement. Absolute maximum score is 10 times number of features used in calculating specificity score. For 5 features, maximum score is 50. Hence threshold can be set a front. Our experience shows that threshold of 40 performs well for creating shorter summaries.
4. Further, the sentences can be reordered to keep the original sequence in the document. Reordering would improve readability of the summary if desired.

Table 3. Specificity score calculation

```
for each record $r_i$=1…n(r) do
      form sentences
       for each sentence $s_i$=1…n(s) do
      POS tagg the sentence $s_i$
      Tokenize the sentence $s_i$
            for each token $t_k$=1…n(t) do
                  If ($t_k$ is not a stopword)
                        then added as a valid token
                        Identify semantic depth of $t_k$
                        Identify semantic height of $t_k$
                        Identify whether it is a NE $t_k$
                        Identify total occurrence count of $t_k$
                        Identify whether $t_k$ is a Proper noun
                   end if
            end for //for tokens
            aggregate the average semantic depth, average semantic height, NE count, average total occurrence count, Sentence length, Number of proper nouns of the sentence $s_i$
            Based upon the above aggregated values identified, calculate the specific score of the sentence
      end for //for sentences
end for//for records
Store sentences according to the specificity in descending order.
Extract the top % of sentences to represent as summary
```

4 Experiment Results and Discussion

4.1 News Articles



a b

Fig. 1. Sentence number vs specificity score a) for news article d01a/SJMN91-06290185, b) average specificity score from 50 news articles (DUC 2001)

The Fig. 1 shows how specificity score vary with sentence order (number). The chart appropriately models intentional structure of news article: The example news article can be divided into ~8 blocks (Fig. 1a, a, b, c, d *etc.*) – each block covering sentences from high specificity score to the next sentence with high specificity score (excluding the next peak). The lead sentences are marked by the sentences with high specificity score and subsequent sentences providing further background in the context of leads are indicated by gradually decreasing specificity score. Interestingly, all the lead sentences are having very high specificity score. Close inspection of these sentences (Table 4) reveals that these sentences introduce sub stories around the main story which is introduced by first sentence.

Table 4. Lead sentences and their specificity score (sco.) for SJMN91-06290185

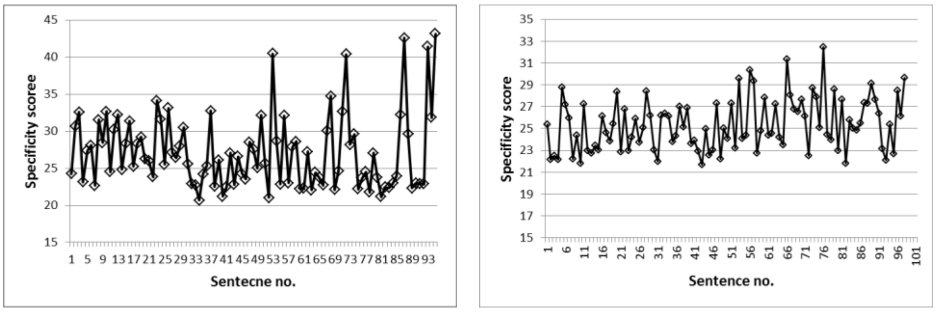
Sentence	sco.
Clarence Thomas, triumphing over eleventh-hour charges of sexual harassment, won Senate confirmation by only four votes Tuesday night to become the youngest member of the Supreme Court and its first black conservative.	43.46
It was the closest Senate confirmation of a Supreme Court nominee since Lucius Q. C. Lamar, an appointee of President Grover Cleveland, also squeezed through by four votes in 1888.	44.36
“Today the Senate sacrificed the integrity of the Supreme Court, its own reputation and the rights of American women to the Bush-Reagan agenda," the Women's Legal Defense Fund said in a statement released after the vote.	42.95
Law Professor Anita Hill, once an aide to Thomas, declined to comment specifically about the Senate vote	41.32

Model summary of this article contains only one sentence from the above list that is 1st sentence and other supporting sentences for this lead sentence. The specificity score of this sentence is second highest - second to the sentence number 6 ("*It was the closest Senate confirmation of a Supreme Court nominee since Lucius Q.C. Lamar, an appointee of President Grover Cleveland, also squeezed through by four votes in 1888*"). One can argue in this case – the very reason the Clarence Thomas wining senate confirmation is a news (apart from its own merit) because it was similar to earlier event (someone winning by 4 votes).

In one embodiment summary can have all the lead sentences without any supporting sentences (as reflected in news story structure). The example (Table 4) can be one of such summary that picks up first *k=4* sentences above some threshold specificity score *S=40*. Variety of such algorithms can be devised very easily with specificity score as the parameter. Current analysis of specificity score for summarization is performed using 5 features (excludes length) and the absolute maximum score a sentence can have is 50. Hence setting up threshold a front or fine turning it for the given set of news stories won't be very difficult. News articles (DUC 2001) about other types of events show similar structure (Fig. 1b). Further comprehensive exploration is required to cover all other types of news.

4.2 Technical Articles

Summary of technical articles also can be seen from specificity perspective. Whereas news article exhibit multiple alternatives of choosing sentences using specificity score (4.1), technical article summary might have limited options. Most of the time summary of technical article is expected to have only specific or only generic sentences. Generally, reader of the technical articles is interested in details such as results, conclusion of the work *etc.* For example, we want to create a bulleted list of research findings. Such expectation from the sentences of a summary can be expressed using high specificity scores directly and hence can be used for generating summary of a technical document. Same is true for technical reports. We begin our investigation with structure of the document as revealed by specificity score.



a b

Fig. 2. Specificity score vs sentence number for a) [2] b) average from [**2**], [**18**], [**19**], [**20**], [**21**]

Structure of the document – in terms of information imparted to the reader – is nicely revealed by the specificity scores. One can easily identify smaller blocks of sentences beginning with low specificity scores and then ending with high specificity score sentences. This is natural given that the technical article would begin with introducing topics to the reader with increasing details as we go on. In this case structure indicated by specificity score overlaps with paragraph structure and can be expected to be a trend for the other technical articles too (Fig. 2b). The difference in technical article and news article is: each block in technical article begin with more generic information and ends with sentences with more and more specific information (further examination with larger dataset is required for higher confidence on this observation, especially for news articles if they begin with more specific sentences). With increase in sentence number specificity score also increases – that is subsequent paragraphs are providing more specific information than the beginning paragraphs. Considering the difference between studies of two types of documents (that is news, and scientific publications), we are tempted to suggest that specificity score can be effectively utilized for identifying genre (technical/news *etc.*) of the document too.

Table 5 shows summary of Relative Utility paper by Radev et. al. [**2**] generated by selecting 10% of sentences. Table 6 shows first 5 sentences of summary of same article generated by lexical chaining algorithm [**20**]. Note that sentences are not reordered according to the original sequence.

Table 5. Summary of paper by Radev *et. al.* [**2**] using specificity score (sco.)

Sentences	Sco.
Second, MEAD and WEBSUM score approximately the same on all metrics with MEAD doing slightly better on the Word overlap , Bigram overlap , and Longest common subsequence measures and WEBSUM on the cosine metric.	42.60
We used the Hong Kong News summary corpus created at Johns Hopkins University in 2001.	40.45
Third , even though the performances of MEAD and WEBSUM S also increase with summary length , MEAD normalized version D decreases slowly with summary length until the two summarizers score about the same on both S and D for longer summaries.	40.36
The single document results tables compare MEAD with WEBSUM and the two baselines RAND and LEAD.	34.72
In fact , the interjudge agreement as measured by RU for this example is0. 76. RU agreement see next section is defined as the relative score that one judge would get given his own extract and the other judge sentence judgements.	34.12
A summarizer could have an RU agreement with judge J1 as low as 0.14 and an agreement with judge J2 as low as0. 38. In other words , given that interjudge agreement is significantly less than 1.0 but significantly more than the worst score possible , an automatic summarizer might score as low as .70 and still be almost as good as the judges themselves.	33.13
Using metrics such as P&R or PA [1 , 2] to evaluate summaries creates the possibility that two equally good extracts are judged very differently.	32.67
The average value of R across all documents at the 5% target length is 0.598 while the average value of J is0. 799. The corresponding values for the 20% target length are R = 0 635 and J = 0 835.	32.66

We will address some advantages of RU over existing co selection metrics such as Precision , Recall , percent agreement , and Kappa.	32.57
Using P&R or PA , system A will be ranked much higher than systemB. It is quite possible however , that for the purpose of summarization , sentences 2 and 3 are equally important , in which case the two systems should get the same score.	32.24

Table 6. First 5 sentences of lexical chaining summary of the RU paper

The main problem with traditional co-selection metrics (thus named because they measure the degree of overlap between the list of sentences selected by a judge and an automatically produced extract) such as Precision, Recall, and Percent Agreement for evaluating extractive summarizers is that human judges often disagree about which the top n% most important sentences in a document or cluster are and yet, there appears to be an implicit importance value for all sentences which is judge-independent.

These include word based cosine between two summaries, word overlap, bigram overlap, and LCS (longest common subsequence). These metrics are all based on the actual text of the extracts (unlike P/R/Kappa/RU, which are all computed on the sentence co-selection vectors).

In the formula for U0, "j (multi-judge summary characteristic function) is 1 for the top e sentences according to the sum of utility scores from all judges.

Relative Utility provides an intuitive mechanism which takes into account the fact that even though human judges may disagree on exactly which sentences belong in a summary, they tend to agree on the overall salience of each sentence.

The Relative Utility (RU) method [3] allows ideal summaries to consist of sentence sets with variable membership.

Table 7. Summary of paper by *Helteran et. al.* [**1**] and specificty score (sco.)

Sentences	Sco.
Some of the generalisation links are part of 3- or 4-link hierarchies, e.g. "FV40 Victim outspoken about/campaigning on immigration issues" (26 mentions) to "FV41 Victim was anti immigration" (23) to "FV42 Victim wanted to close borders to immigration" (9), or "FV50 Victim outspoken about race/religion issues" (17 mentions) to "FV51 Victim outspoken about Islam/Muslims" (16) to "FV52 Victim made negative remarks about Islam" (14) to "FV53 Victim called Islam a backward religion" (9).	43.18
In principle , the comparison can be done via coselection of extracted sentences Rath et al. , 1961; Jing et al. , 1998; Zechner , 1996 , by string based surface measures Lin and Hovy , 2002; Saggion et al. , 2002 , or by subjective judgements of the amount of information overlap DUC , 2002 .	40.00
In the past years, there has been quite a lot of summarisation work that has effectively aimed at finding viable evaluation strategies Sparck Jones , 1999; Jing et al. , 1998; Donaway et al. , 2000 .	36.48
The factoid approach can capture much finer shades of meaning differentiations than DUC style information overlap does - in an example from Lin and Hovy (2002), an assessor judged some content overlap between "Thousands of people are feared dead and "3, 000 and perhaps . 5, 000 people have been killed."	36.21
Pim Fortuyn , a Dutch right wing politician , was shot dead at a radio station in Hilversum.	35.67

The text used for the experiment is a BBC report on the killing of the Dutch politician Pim Fortuyn.	35.10
However, Lin and Hovy 2002 report low agreement for two tasks producing the human summaries around 40% , and assigning information overlap between them.	35.02
Largescale conferences like SUMMAC Mani et al., 1999 and DUC 2002 have unfortunately shown weak results in that current evaluation measures could not distinguish between automatic summaries - though they are effective enough to distinguish them from human written summaries.	35.01
In summarisation there appears to be no “one truth”, as is evidenced by a low agreement between humans in producing gold standard summaries by sentence selection Rath et al. , 1961; Jing et al. , 1998; Zechner , 1996 , and low overlap measures between humans when gold standards summaries are created by reformulation in the summarisers' own words e.g. the average overlap for the 542 single document summary pairs in DUC-02 was only about 47% .	34.41
Lin and Hovy 2002 examine the use of a multiple gold standard for summarisation evaluation, and conclude \we need more than one model summary although we cannot estimate how many model summaries are required to achieve reliable automated summary evaluation .	33.50

Table 8. First 5 sentences of lexical chaining summary of *Helteran et. al.* [**1**]

We present a new approach to summary evaluation which combines two novel aspects, namely (a) content comparison between gold standard summary and system summary via factoids, a pseudo-semantic representation based on atomic information units which can be robustly marked in text, and (b) use of a gold standard consensus summary, in our case based on 50 individual summaries of one text.
If we decide to use a single human summary as a gold standard, we in fact assume that this human's choice of important material is acceptable for all other summary users, which it the wrong assumption, as the lack of consensus between the various human summaries shows.
All in all, the use of consensus summaries and factoid analysis, even though expensive to set up for the moment, provides a promising alternative which could well bring us closer to a solution to several problems in summarisation evaluation.
In summarisation there appears to be no \one truth", as is evidenced by a low agreement between humans in producing gold standard summaries by sentence selection (Rath et al, 1961; Jing et al, 1998; Zechner, 1996), and low overlap measures between humans when gold standards summaries are created by reformulation in the summarisers' own words (eg the average overlap for the 542 single document summary pairs in DUC-02 was only about 47%).
3, There is no such thing as overall consensus, but there is a difference in perceived importance between the various factoids, We can determine whether this is the case by examining how often each factoid is used in the summaries, Factoids that are more important ought to be included more often, In that case, it is still possible to create a consensus-like reference summary for any desired summary size.

5 Conclusion

We demonstrated how summarization can be driven by a parameter other than importance. Summary produced by such a method provide mechanism for choosing right sentences as per the users expectation from the summary. Summary using specificity score outperforms (assessed using sample cases) summaries by some popular summarization techniques in case detailed information from summary is expected by more informed reader. Evaluation of such a summary is not possible by existing summary evaluation methods that use model summaries by experts as such summaries tend to provide introductory information. Some of the specific observations are:

- Specificity score per sentence provides easy way to assess the structure of the document from information perspective and could be used further for identifying type of the document.
- Specificity score based approach can create a summary to have detailed or introductory information in the given document by setting a threshold for the score, or using sorted list and then by selecting top k sentences as required. Further, complex strategies for choosing right mix of specific and generic sentences can be devised for appropriate summary: Let's say we have a budget of score S (say 150) and task is to choose right sentences within the limits – one extreme case would be choosing many low scoring sentences and another a few high scoring ones.

Design of summarization system need to consider three context factors namely INPUT, PURPOSE, and OUTPUT[1]. INPUT factors and OUTPUT factors (Material, style, expression *etc.*) characterize input material and output material respectively, and PURPOSE factors (audience, use *etc.*) are related to the usage of summary [**16**], [**22**]. It is natural to think that robust summarization system then needs some parameters characterizing each of the above mentioned factors - mainly purpose and output. Many of these factors influence each other in complex way with varying degree (for example, style and expression, brevity and use). The influence on each other can be leveraged to create system specifications (for summary) with only limited number of factors. Especially in some cases PURPOSE fully determines the OUTPUT [**22**], for example, if the reader is reviewing papers for literature survey then abstract like summary might be fine but if he is more informed reader then he would be more interested in scientific claims, results and so on. Such couplings between context factors are usually reflected in more observable parameters such as discourse relations. For the technical paper example in this paper, specificity controlled the summarization process as one would be looking for more specific information. Thus, specificity shows potential to characterize some of the PURPOSE and/or OUTPUT factors of summary. Consensus on summary factors (and their definitions) is required for further elaborate investigations on how summarization can be driven by factors like specificity.

References

1. Halteren, V., Teufel, S.: Examining the Consensus between Human Summaries: Initial Experiments with Factoid Analysis. HLT-NAACL-DUC '03 Proceedings of the HLT-NAACL 03 on Text summarization workshop (2003)

[1] See DUC roadmap 2005-2007 http://duc.nist.gov/RM0507/rm.html for details

2. Radev, D., Tam, D.: Summarization Evaluation Using Relative Utility. In: Proceeding CIKM '03 Proceedings of the twelfth international conference on Information and knowledge management, p.508/511 (Nov 2003)
3. Lin, C.-Y.: Looking for a Few Good Metrics: Automatic Summarization Evaluation — How Many Samples are Enough? In: NTCIR Workshop 4, Tokyo, Japan (June 2-4, 2004)
4. Nenkova, A.: Summarization Evaluation for Text and Speech: Issues and Approaches. (Accessed on Jan 2015) Available at: http://www.cis.upenn.edu/~nenkova/papers/sumEval.pdf.
5. Nenkova, A., Passonneau, R.: Evaluating Content Selection in Summarization: The Pyramid Method. North American Chapter of the Association for Computational Linguistics - NAACL, 145/152 (2004)
6. Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain (2004)
7. Hovy, E., Lin, C.-Y., Zhou, L., Fukumoto, J.: Automated Summarization Evaluation with Basic Elements. In: The Fifth Conference on Language Resources and Evaluation (LREC) (2006)
8. Louise, A., Nenkova, A.: Text Specificity and Impact on Quality of News Summaries. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, pp.34-42 (2011)
9. Erkan, G., Radev, D.: Lexrank: Graphbased Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 22, 457-479 (2004)
10. Palshikar, G., Deshpande, S., Bhat, S.: Quest: Discovering Insights from Survey Responses. In: Proceedings of 8th Australasian Data Mining Conf. (AusDM09), pp.83-92 (2009)
11. Deshpande, S., Palshikar, G., G, Athiappan.: An Unsupervised Approach to Sentence Classification. In: Proceedings of International Conference on Management of Data", COMAD 2010, Nagpur, India (2010)
12. Louis, A., Nenkova, A.: General versus Specific Sentences: Automatic Identification and Application to Analysis of News Summaries.
13. Jones, K.: A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation 28(1), 11-21 (1972)
14. Hassel, M.: Evaluation of Automatic Text Summarization, Licentiate Thesis., Stockholm, Sweden (2004)
15. Jing, H., McKeown, K.: Cut and Paste Based Text Summarization. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference, Seattle, Washington (2000)
16. Sparck Jones, K.: Automatic Summarizing: Factors and Directions., Cambridge, MA: MIT Press (1999)
17. Fellbaum, C.: WordNet: an On-line Lexical Database and its Applications. MIT Press (1998)
18. Doran, W., Stokes, N., Carthy, J., Dunnion, J.: Comparing Lexical Chain-based Summarisation Approaches Using an Extrinsic Evaluation. (2004)
19. Apte, H.: Building a Trainable Multi-document Summarizer. (Accessed Jan 2015) Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.4063&rep=rep1&type=pdf
20. Palshikar, G., Deshpande, S., Athiappan, G.: Combining Summaries Using Unsupervised Rank Aggregation. Computational Linguistics and Intelligent Text Processing 7182, 378-389 (2012)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp.311-318 (2002)
22. Sparck Jones, K.: Automatic Summarizing: The State of the Art. (2007)

Readable and Coherent MultiDocument Summarization

Litton J. Kurisinkel,[1] Vigneshwaran M.,[1] Vasudeva Varma,[2] Dipti Misra Sharma [3]

International Institute of Information Technology / Hyderabad 500032, India
litton.jKurisinkel@research.iiit.ac.in, vigneshwaran.m@research.iiit.ac.in
vv@iiit.ac.in, dipti@iiit.ac.in

Abstract. Extractive summarization is the process of precisely choosing a set of sentences from a corpus which can actually be a representative of the original corpus in a limited space. In addition to exhibiting a good content coverage, the final summary should be readable as well as structurally and topically coherent. In this paper we present a holistic, multi-document summarization approach which takes care of the content coverage, sentence ordering, maintenance of topical coherence, topical order and inter-sentence structural relationships. To achieve this we have introduced a novel concept of a Local Coherent Unit(LCU). Our results are comparable with the peer systems for content coverage and sentence ordering measured in terms of ROUGE and τ score respectively. The human evaluation preference for readability and coherence of summary are significantly better for our approach vis a vis other approaches. The approach is scalable to bigger real-time corpus as well.

1 Introduction

Automated text summarization enables the reader of the summary to understand the essence of information contained in a big corpus of documents without going through the entire set. Extractive summarization techniques try to achieve this by selecting a proper subset of sentences from the corpus, which constitute the summary. Most of the techniques adopted for extractive summarization can be understood to perform three basic steps.

1. Create an intermediate representation for the target text such that the key textual features within are captured.
2. Using the generated intermediate representation, assign scores for individual sentences within the text.
3. Finally select a set of sentences which maximizes the total score as the summary of the target text.

Possible intermediate representations are created by Topic Signatures, Word frequency count approaches, Latent Space Approaches using matrix factorization or Bayesian Approaches. In almost all of the approaches the smallest linguistic unit which is to be scored and selected for summarization is a sentence. Most of the prevalent scoring functions consider quantifying the priority of the sentence for better content coverage. In these approaches, the output set of sentences are later fed to a distinct sentence-ordering component which reorders the sentences. By the time a precise subset of sentences are

pp. 295–311; rec. 2015-01-22; acc. 2015-02-25

chosen, most of the information related to the inter-sentential structural dependency are lost. Most of the re-ordering algorithms can achieve only a topical order leaving behind the possibility of out-of-context sentence usage as given below.

> *e.g. Nevertheless this object pulls everything which enters its event horizon.*

The above sentence might secure a high score in terms of topical significance if the corpus is on *Black hole* but can still result in an out-of-context sentence placement. This can cause an incoherent reading or sometimes result in an erroneous inference. We propose a novel concept called as a Local Coherent Unit(LCU) which enforces a contextual constraint for sentence extraction. *An LCU is a unit of text containing sequence of sentences such that, excluding the first sentence, every subsequent sentence within the unit has an explicit discourse dependency with the preceding sentence.* The explicit discourse dependency can be of any type such as an event adverbial related to previous sentence, anaphoric reference, deictic pointers to previous entities etc. These are realized in sentences as structural dependency cues.

We discuss about the relevant works done on summarization and sentence ordering in Section 2. In section 3 we discuss an overview of all the components of our system and their organization. The section comprises of the subsection 3.1 which explains about a stand-alone component which identifies LCUs. Then we elaborate about the topic modelling, document merging, topic segmentation in subsections 3.2, 3.3 and 3.4 respectively. The role of topic segmentation and LCU in our summarization process is explained in subsections 3.5 and 3.6. Finally the experimental results are discussed in section 4.

2 Related Work

Extensive work has been done on extractive summarization which tries to achieve a proper content coverage by scoring and selection of sentences. All these previous works seek the help of a second component to re-order the set of extracted sentences. Most of the extractive summarization researches aim to increase the total salience of the sentences while reducing redundancy. Approaches include the use of Maximum Marginal Relevance [1], Centroid-based Summarization [2], Summarization through Keyphrase Extraction [3] and Formulation as Minimum Dominating Set problem [4]. Graph centrality has also been used to estimate the salience of a sentence [5]. Approaches to content analysis include generative topic models [6], [7],[8] and Discriminative models [9].

ILP2 [10] is a system that uses Integer Linear Programming(ILP) to jointly optimize the importance of the summary's sentences and their diversity (non-redundancy), while also respecting the maximum allowed summary length. They use a Support Vector Regression model to generate a scoring function for the sentences. Woodsend and Lapata [11] arrived at a scoring function which holds linear components to quantify the salience of bi-grams, salience of parse tree nodes and a component based on a language model which penalises the unlikely sentences. An approach based on the distribution of some important concepts in the summary was done by [12]. The concepts are bi-grams in the corpus to be summarised. They formulated an ILP objective function in the space of candidate summaries that maximizes the total concept weight score of the summary to be chosen.

Takamura and Okumura [13] have treated multidocument summarization as a maximum concept coverage problem with knapsack constraint(MCKP). They have also exploited the possibility of decoding algorithms in solving MCKP in the summarization task. Lin and Bilmes[14] formulated summarization as a sub-modular function maximization problem in the possible set of candidate summaries with due respect to the space constraint. All the above methods have concentrated on content coverage but have the drawback of out-of-context sentence usages.

As far as sentence ordering is concerned, Li et al. [15] used context inference to achieve better sentence ordering while McKeown et al[16] used majority ordering algorithm to sort sentences. Lapata [17] provided an unsupervised probabilistic model for sentence ordering while Ji et al [18] used a cluster adjacency based approach. Though the sentence ordering approaches can achieve a topical order of sentences, the local structural relations of the sentences are never captured.

The work which pioneered a holistic approach towards multi-document summarization by bringing sentence selection and coherence under a single umbrella is G-Flow by [19]. They built a graph which stored discourse relations with proper edge weights to quantify coherence. This value was linearly combined along with salience and redundancy in the scoring function of sentences to formulate multi-document summarization as a constraint optimization problem.

The system has taken into consideration the readability of the extracted sentences in output summary by quantifying its coherence by means of discourse graph. With the increase in corpus size, the space complexity of generating discourse graph with large ‘n’ is of the order O(n^2). The optimization function in this case cannot take a greedy approach for inducing coherence while selecting and discarding sentences for output summary. This is because the coherence is measured for the whole chosen candidate summary and there is no way to greedily choose potentially coherent sentences individually. As per [14], having the objective function as a submodular non-decreasing function can incorporate a greedy approach that guarantees a solution at the most as good as the best solution with a factor of 0.632. Hence we have used an LCU based submodular non-decreasing function in our summary extraction step while letting LCU ensure the required readability and coherence.

3 Our Approach

More than sentence scoring and content selection which aim only at content coverage, a summary should be readable and intelligible to a human reader without any previous knowledge on the content of the corpus. A summary which is topically uniform may not capture different topical aspects of the corpus and a summary which is too diverse can take the form of a short note which can only be understood by a person having prior knowledge about the content of the corpus. So an optimal topical coherence, which conveys a gist of the various topics of the corpus, packed within the constraints of target summary size along with a proper sentence order needs to be achieved.

The main intuition behind the choice of our approach begins with a crucial question about the linguistic nature of a text. *Is text a bag of words every time?* It need not be so because, for instance, the first sentence taken from each paragraph of a document can account for a reliable summary of the whole document. Psycholinguistic studies suggest that local coherence plays a vital role in inference making during the reading [20]. Local coherence is undoubtedly necessary for global coherence and has received considerable attention in Computational Linguistics. (Marcu [21], Kintsch et al[22], Althaus et al[23], Karamanis et al [24]).

To handle the explicit structural coherence created by the sentences in a document, we conceptualized a notion called as Local Coherent Unit(LCU). An LCU is a unit of text containing a sequence of structurally dependent adjacent sentences in a document. The LCU will be used as a basic unit of processing for summary extraction which implicitly imposes restriction of out-of-context sentence usage and hence more readable. The next section describes in more detail what is meant by structural dependency between sentences, how to identify Local Coherent Units etc.

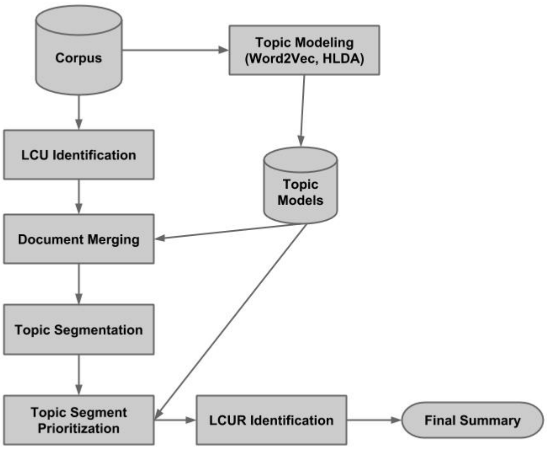


Fig. 1. System Architecture

The architecture of the system is shown above. A brief explanation of the above system architecture is as follows:

1. Given a corpus of multiple documents, a complete, non-overlapping set of LCUs in every one of those documents are identified. Now the basic unit of processing for every document is a sequence of LCUs, not sentences.
2. A HLDA tree which represents the latent topic structure based on term distribution is created for the entire corpus by considering paragraphs as documents i.e the entire set of paragraphs is the input corpus for HLDA
3. Word2Vec tool takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The word vector so generated contains top ‘n’ list of

words closer to the current word based on cosine distance between them. Such a Word2Vec model is created for the entire corpus.

4. Word2Vec vectors of all the terms in an LCU are calculated. Now the mean of all the word vectors in the LCU is considered to represent the LCU itself in meaning space. Any two LCUs can be compared for topical similarity using cosine similarity of their mean word vectors. Two LCUs can likewise be compared for parent-child hierarchy of their terms using HLDA tree.
5. Using the above topic models, a total topical ordering of the LCUs in the entire corpus is performed. We call this step as document merging. The merged document now is the single output containing sequence of all LCUs in the corpus in topical order.
6. A topic segmentation is performed explicitly on the merged document to identify topic boundaries thus creating segments of topics. Again a topic segment is just a sequence of LCUs in the merged order since we have not disturbed it in topic segmentation. This step is required on top of the merged document in order to scale up the coherent summary extraction approach for bigger corpus with multiple documents containing larger text.
7. Final summary is something that has to be extracted from diverse non-redundant topics while the sentences extracted have to be readable in the sequence of extraction.
8. For this, we find topic priority of each topic segment to identify its contribution to the final summary. Finally noise-free representatives of LCU are chosen from every topic segment proportional to their priority such that the extracted summary is optimal.

3.1 Local Coherent Units Identification

Every document in a corpus is a set of sentences which together form a discourse. For summarization it is necessary to retain the discourse level relations between sentences and make use of those while extracting content for summary. Typically discourse level relations can be identified by a discourse parser developed based on Rhetorical Structure Theory[25], Penn Discourse Tree Bank[26]. Usefulness of discourse indicators for content selection in summarization has been explored [27] and the robustness of structure information in the identification of importance of a text has been discussed.

However we observed that instead of explicitly modelling the discourse relations between sentences, representation of document as a set of Local Coherent Units helps capture the inter-sentence structural dependencies that can be best utilized for summary extraction incrementally. The structural dependency is defined in terms of a set of linguistic cues obtained from the dependency tree for every individual sentences. The identification of these local coherent units can also be done for languages which do not have a fully developed discourse parser and hence we decided to apply this strategy as a component in multi-document summarization.

Initially we start with one empty LCU. Once we have the parsed output of the current document from a dependency parser[1], the decision that has to be taken for each

[1] Stanford Dependency parser Version 3.3.1,
http://nlp.stanford.edu/software/lex-parser.shtml#Download

Algorithm 1 LCU Identification Algorithm

```
Require: $DEP\_PARSE\_TREES$, $DEP\_CUES$, $ARG\_CUES$
  for each i $\in$ $DEP\_PARSE\_TREES$ in the document do
    $cue\_dep\_stack$ $\leftarrow$ "ROOT"
    $cue\_arg\_stack$ $\leftarrow$ ""
    flag $\leftarrow$ 0
    while $cue\_dep\_stack \neq$ "" do
      temp $\leftarrow$ $cue\_dep\_stack$.pop()
      $cue\_arg\_stack$ $\leftarrow$ searchRightArg(i,temp)
      while $cue\_arg\_stack \neq$ "" do
        temp2 $\leftarrow$ $cue\_arg\_stack$.pop()
        if temp2 is an entry in $DEP\_CUES$ then
          flag $\leftarrow$ 1
          Break from immediate while loop which checks $cue\_arg\_stack$
        else
          $cue\_arg\_stack$.push(temp2)
        end if
        if flag==1 then
          ADD THE CURRENT SENTENCE i TO EXISTING LCU
        else
          ADD THE CURRENT SENTENCE i TO NEW LCU
        end if
      end while
    end while
  end for
```

sentence is whether it belongs to a previous LCU or begins a new one. Some linguistic cues('nsubj' modified by demonstratives etc.) were used to decide whether the current sentence has a structural dependency on previous sentence. If such a dependence exists the current sentence is added to the existing LCU. If not, a new local coherent unit is created and the sentence is added to it. This is continued till the end of the document and as a result the document will be segmented as a series of LCUs. By processing all the documents of the corpus in the same manner, we get a representation now where the documents are understood as a series of LCUs which can be used later for applying statistical methods.

Example for a local coherent unit is given below.

> e.g. *Black holes are intriguing ideas.* However *they are not likely to account for much dark matter.*

The above LCU has two sentences in it. *However* is a CC in the main clause of the second sentence which shows structural dependency with first sentence. By a simple set of rules which take the cues tabulated in Table 3.1 and using a finite set of arguments for such dependency relations mentioned below LCU can be formed.

The dependency parsed output of an entire document is taken as the input and the Algorithm 1 is run to get the sequence of LCUs identified within the document. For executing the algorithm we need set of cue dependencies and a set of cue argument values

Algorithm 2 Search Right Arguments Recursively

```
Require: $DEP\_TREE, CURR\_DEP\_RELATION$
  $arg\_stack \leftarrow \{\}$
  for All lines throughout the parse tree do
    if Right argument of $CURR\_DEP\_RELATION$ occurs as Left argument in one of $DEP\_CUES$ then
      Add the Right argument to $arg\_stack$
    end if
  end for
  return $arg\_stack$
```

Table 1. Cue dependencies called as DEP_CUES for LCU Identification

Relation	Meaning	Examples
nsubj	Subject of main clause	He,She
dobj	Direct object of main verb	He,They
det	Determiner,Demonstratives	This,The
mark	Subordinate marker	that, if
nsubjpass	Subject of passive	He,It
advmod	Adverbial modifier	Still, thus
CC	Coordinate conjunction	And, Yet

which trigger structural dependency with preceding sentence are finite in number. We have chosen the below list of dependencies and arguments.

1. Dependency cues DEP_CUES *root, nsubj, dobj, det, mark, nsubjpass, advmod, cc*
2. Argument cues ARG_CUES - *All third person pronouns and their inflected forms, Demonstratives, 20 adverbs which act as explicit discourse connectives such as so, thus, still etc*

There are many discourse markers and information structure cues English. But as a preliminary approach we have chosen the above ones as these exhibited a reasonable coverage of correct LCU identification as shown in 4. The approach can be extended by using other discourse relations as well. The dependency parse tree of a line consists of entries of the form Dependency_Reln(leftarg, rightarg). The Algorithm 2 tries to find recursively all right arguments which are present as left argument for some cue dependency relations in the parse tree. All such identified right arguments are added to the stack cue_arg_stack and returned.

3.2 Word2Vec and HLDA Modelling

Reliable topic models created for the corpus can enhance the process of automated summarization. Topic Hierarchy of the corpus is identified by creating an HLDA model[2] [28]. The paragraphs in a document hold the explicit topic-wise organization of text

[2] https://github.com/chyikwei/topicModels

conceived by the author. So the paragraphs hold a sufficient amount of prior information about the topic-term distribution. Therefore for the purpose of HLDA tree creation, we treat paragraphs as documents and the entire set of paragraphs as input corpus for HLDA. As paragraphs are usually fine grained on a few topics in a well written document, variable α for HLDA which corresponds to the prior for per document(paragraph) topic distribution is kept at a very low value. We have created a Word2Vec model[3] to find the semantic similarity between any two text units. Each word is vectorised by choosing top 'n' similar words from Word2Vec and their corresponding similarity values. *To vectorise a text unit we take the mean vector of all word vectors in the text unit.*

3.3 Document Merging

Once each document is represented as a sequence of LCUs, each LCU in the corpus is assigned a corpus level Id . Local coherent units are relatively much larger than a sentence and hold enough information to decide their topical identity. The task of summarization becomes easier once we could merge these documents into a topically coherent document without violating inter-sentence structural relationships. As LCUs already hold inter-sentence structural relationships, arriving at a sequence of corpus-level LCU ids which exhibits maximum topical order and coherence can result in the best merged document that is possible. *For this purpose we utilise the HLDA and Word2Vec model created for the corpus.* Document merging can be framed as an optimization problem where we maximize the function given by the Equation (1) in the space of all possible sequences of LCUs.

$$\begin{aligned} \mathbf{Q}(\mathbf{Z}) = \sum_{i=1}^{N-1} \mathbf{100} * \mathbf{W2V}(\mathrm{LCU}_i, \mathrm{LCU}_{i+1}) + \\ \mathbf{DD}(\mathrm{LCU}_i, \mathrm{LCU}_{i+1}) \; - \; \mathbf{DD}(\mathrm{LCU}_{i+1}, \mathrm{LCU}_i) \end{aligned} \tag{1}$$

where

Z→ *Possible sequence of LCUs in the corpus*
N → *Total number of LCUs in the corpus.*
W2V → *Word2Vec cosine similarity*
$\mathrm{LCU_i}$ → *i^{th} LCU in the sequence.*
DD → *Function call to Algorithm4*

Algorithm(4) quantifies the extent upto which the topics dealt in the LCU2 belong to the sub-topic category of LCU1. First term in the above Equation (1) brings all coherent units which deal with semantically similar topics together. Second and third terms arrange them in a proper topic to subtopic order. Since the framing of document merging as an optimization problem can be costly for real-time usage, we have used a greedy algorithm which approximates the function in the Equation 1. For the convenience of

[3] http://radimrehurek.com/gensim/models/Word2Vec.html

greedy approximation at each step we have refreamed the 2 as below.

$$\mathbf{F}(\mathbf{LCU1},\mathbf{LCU2}) = \mathbf{100} * \mathbf{W2V}(\mathrm{LCU}_1,\mathrm{LCU}_2) + \mathbf{DD}(\mathrm{LCU}_1,\mathrm{LCU}_2) \; - \; \mathbf{DD}(\mathrm{LCU}_2,\mathrm{LCU}_1) \tag{2}$$

Algorithm 3 Document Merging Algorithm

```
Require: Doc1, Doc2, HLDA model
  maxScoringPair ← (0,0)
  currentFnValue ← 0
  maxvalue ← 0
  for each LCU i ϵ Doc2 do
    for each LCU j ϵ Doc1 do
      currentFnValue ← F(i,j)
      if currentFnValue > maxvalue then
        maxScoringPair ← (i,j)
        maxvalue ← currentFnValue
      end if
      currentFnValue ← F(j,i)
      if currentFnValue > maxvalue then
        maxScoringPair ← (j,i)
        maxvalue ← currentFnValue
      end if
    end for
    if maxScoringPair = (i,j) then
      insert i above j in Doc1
    end if
    if maxScoringPair = (j,i) then
      insert i below j in Doc1
    end if
  end for
  return Doc1
```

The Algorithm(3) uses the above function F in Equation(2). It takes two documents, arranges the LCUs from two documents in the optimum order and returns the merged document. In the algorithm 4 *H* stands for the height of HLDA tree, *x.level* is the level of topic node in the HLDA tree to which the term x belongs with a maximum chance, AncestorNodes(i) is the set of all ancestor nodes of the node in the HLDA tree to which term i belongs with maximum chance, DescendentNodes(i) is the set of all descendent nodes of the node in the HLDA tree to which the term i belongs with maximum chance.

Overall we are trying to find the insertion position of coherent unit in the document which maximizes the above function given by Equation 2. The merge algorithm starts with first two documents to form a single merged document. This merged output is further merged with the third document and the process incrementally continues until all the documents in the corpus are merged into a single structure.

Algorithm 4 Descendent Level Difference Calculation

```
Require: LCU1,LCU2,HLDA model
  levelDiff ← 0
  for each term i ϵ LCU1 do
    for each term j ϵ LCU2 do
      if j ϵ Descendents(i) then
        levelDiff ← levelDiff+H-(j.level-i.level)
      end if
    end for
  end for
  return levelDiff
```

3.4 Linear Topic Segmentation Using Affinity Propagation Algorithm

The larger merged document formed as a sequence of all local coherent units in the corpus is linearly segmented into topic segments which contain more than one local coherent units. Each topic segment exhibits a high level of topic uniformity. We employ the implementation of ‘Linear text segmentation by affinity propagation’ by [29] for segmenting the merged document.

Affinity propagation algorithm for segmentation receives a set of pairwise similarities between data points and decides the topic segment boundaries and segment centres. A segment centre is a data point which best describes all other data points within the segment. Data points in our merged document are local coherent units. The similarity measure to be supplied to the topic segmentation algorithm is calculated by the cosine similarity between mean word vector of local coherent units. Another important parameter for topic segmentation algorithm is the set of preference values which represents the *a priori* belief of each data point to become a segment centre. Preference value of a local coherent unit during linear topic segmentation is calculated as the mean cosine similarity between k neighbouring local coherent units in the merged document[4].

3.5 Prioritization of each Topic Segment for Summarization Process

As an analogy to understand the topic segments, it can be seen that reduction of an image from a richer dimension to lower dimensions can cause certain objects in the image to get eliminated and some among them to get abstracted. Topic segments in the merged document are analogous to the objects in the high resolution image. We prioritize the topic segments and identify their level of participation in the final summary. During this process some among them get abstracted and some get eliminated to generate a coherent summary that is best conveyable within the allowed summary space. The priority of the topic segment *T* is decided by the Equation *(3)* below.

$$\mathbf{P(T)} = \omega 1 * \mathbf{SDI(T)} + \omega 2 * \mathbf{G(T)} \tag{3}$$

where P(T) refers to the priority of the topic segment T, SDI(T) refers to Shannon's diversity index of the topic segment T and G(T) refers to the generality of the topic segment.

[4] The value of ’k’ is experimentally optimized.

The first component decides the information content of the topic segment where the second one decides the generality of information contained. The two terms SDI Shannon's diversity index and Generality are given by

$$\mathrm{SDI(T)} \ = \sum \mathrm{p}_i \ln(\mathrm{p}_i) \ \forall \, \mathrm{term\ i} \in \mathrm{T} \tag{4}$$

where $\mathrm{p_i}$ is the normalized frequency of term *i* in *T*.

$$\mathrm{G(T)} = \sum \frac{(\mathrm{H} - (\mathrm{t.level})}{\mathrm{n} * \mathrm{H}} \\ \forall \, \mathrm{terms\ t} \ \in \ \mathrm{topic\ segment\ T} \tag{5}$$

where *H* is the height of HLDA tree, *n* is the total number of terms in topic segment *T*. The segment priorities calculated from Equation (3) are normalized between 0 and 1. The proportional contribution *binsize$_i$* of each topic segment *i* for the final summary is calculated as

$$\mathrm{binsize_i} \ = \ \mathrm{P(i)} * \mathrm{Targeted\ summary\ size} \tag{6}$$

3.6 Summarization of each Topic Segment

Topic segments are summarized by selecting noise-free representatives of LCUs till the allotted bin size of the topic segment is exhausted. These noise-free representatives are called LCURs.

We have identified Local Coherent Units to avoid an out-of-context sentence usage. But when a non-pruned local coherent unit which is relatively larger than a sentence is directly included as a representative in the final write-up, it can result in a noisy summary in terms of relevance and generality. We have to extract a noise-free combined representative of a local coherent unit without disturbing the structural dependency that is preserved within an LCU.

For this purpose we consider that every sentence in an LCU is depending on all of its previous sentences. So the possible candidate representatives of an LCU containing sentences S1, S2, S3 and S4 are {S1}, {S1,S2}, {S1,S2,S3}, {S1,S2,S3,S4} and we call them local coherent unit representatives(LCUR). This combination was currently chosen to ensure that even after an LCU is pruned the structural dependency between the resulting sentences should be retained. If we choose a combination such as S1,S3 from within an LCU, the possible structural dependency of sentence S3 with S2 would be lost. Such pruning would defeat the purpose of having an LCU.

We use a variant of greedy version of Maximum Word Coverage Algorithm[13] for summarization. We greedily choose the LCUR with highest normalized score of Equation (7) as the candidate representative of the given LCU. At any given instance if a candidate LCUR is chosen, all other LCURs from the same local coherent unit are discarded.

$$\mathrm{SF(LCUR)} = \lambda 1 * \sum \frac{(\mathrm{TF\text{-}IDF(w_i)})}{\mathrm{n}} + \\ \lambda 2 * \mu + \lambda 3 * (\mathrm{TZ}) + \lambda 4 * (\mathrm{FZ}); \tag{7}$$

where

$W_{set} \rightarrow$ *set of words chosen so far in summary at the current iteration of greedy algorithm.*
Word $w_i \in$ LCUR and $w_i \notin W_{set}$.
$n \rightarrow$ Number of words in LCUR.
$\mu \rightarrow$ Average size of sentences in LCUR

$TZ \rightarrow \frac{\text{Size of LCU to which LCUR belongs}}{\text{Topic segment size}}$.
$FZ \rightarrow \frac{\text{Bin size allotted for the topic segment}}{\text{SizeofLCUR}}$

Topic segments are treated as documents to calculate TF-IDF of words w_i. The second term in Equation (7) gives a small priority for the LCURs containing longer sentences while the third term in the function gives a slight priority for an LCUR of a longer LCU. FZ term encourages the selection of LCURs from different LCUs. In each iteration the Greedy algorithm selects the maximum scoring LCUR and continues to include them till the topic segment summary *equals or just crosses the segment's allotted bin size*. We do not form topic segment summaries less than the allotted bin size in order to avoid an aggregate deficit in the targeted summary.

We repeat this for all topic segments and aggregate the set of LCURs from each of the topic segment summaries. The total size of the summary formed out of these bunch of aggregated LCURs could slightly exceed the targeted summary length(Because for every topic segment we chose *topic summary of length $>=$ binsize*). Now the same greedy algorithm with the above mentioned scoring function is applied on the aggregated LCURs such that the final summary does not exceed the targeted summary size in number of bytes.

In the objective function used above, the average TF-IDF score for an LCUR is calculated only for the words which are uncovered by the summary till the current iteration of greedy algorithm. This avoids the explicit usage of diversity measure. As this component of the function is submodular and non-decreasing and all other components have constant values for an LCUR at any stage of iteration, the function SF is submodular and non-decreasing. The LCURs extracted are arranged in the same order in which they occur in the merged document.

4 Experiments and Results

Different components of the system such as Local Coherent Unit Identification, Document Merging, Sentence Ordering and Content Coverage are evaluated using DUC 2004 Task2 Dataset [5] as it contains documents of sufficient size for HLDA modelling[28]. As proper sentence ordering is a consequent of document merging, both need not be tested separately. DUC 2004 contains 50 cluster of documents each containing 10 documents and 4 manual summaries.

4.1 Content Coverage

We have taken DUC 2003 as our development set on which function weights of Equation (3) and Equation (7) and HLDA parameters are optimized using grid search. HLDA

[5] http://www-nlpir.nist.gov/projects/duc/data/2004_data.html

parameters α, β and γ are optimized for achieving better ordering of merged document for each cluster in terms of Kendall's τ (Lebanon, 2002) measure.The weights of Equations (3) and (7) are optimized for achieving maximum ROUGE score [30] with reference summaries. The major systems which has reported results on DUC 2004 dataset for *Content coverage* are [31] [13], [14] and G-FLOW [19]. We have chosen domain independent generic features for summarization and got comparable results in terms of ROUGE-1 recall and F-measure values. We have tested the system with and without topic segmentation. Results of content coverage are tabulated in Table 2.

Table 2. Content Coverage Results

Approach	Rouge-R	Rouge-F
Nobata&Sekine(2004)	30.44	34.36
G-FLow(2013)	37.33	37.43
Our system (Without Topic Segmentation)	37.65	37.70
Our system (With Topic Segmentation)	36.42	36.65
Takamura&Okumura (2009)	38.50	-
Lin & Bilmes(2011)	39.35	38.90

When the system was tested without employing the topic segmentation thereby treating the whole merged document as one topic segment, the content coverage was high but readability and coherence was relatively lesser. With topic segmentation, it can be seen that the content coverage was comparable while at the same time sentence ordering is improved.

4.2 Sentence Ordering

In addition to content coverage, we have compared the results of our approach with the results of existing sentence ordering approaches of [15], [16], [17] and [18].

As the reference summaries of DUC 2004 Task2 contained human framed sentences for each sentence we have chosen the offset of LCU in the merged document which has maximum cosine semantic similarity with the sentence to represent its position with respect to our system. Offsets of sentences in a reference summary has to be in increasing order. The difference of the actual order with a desired increasing order is measured using Kendall's τ. Our average measure for the corpus is comparable with other peer systems for sentence ordering. The results are tabulated in table 3.

4.3 LCU Identification Accuracy

To find out how accurate the identified Local Coherent Units are, we compare it against the manually identified local coherent units on the test corpus. We had chosen a collection of 5 sample documents from DUC corpus for which we had manually identified the Local coherent units. The percentage accuracy of proper identification of local coherent unit is measured as the number of edit operations required to align the system output

Table 3. Sentence Ordering Results

Approach	Kendall's τ
McKeown et al.	0.143
Lapata et al.	0.144
Our System	0.387
Ji et al.	0.415
Li et al.	0.432

with the ideal manual annotation of LCUs. The number of sentences moved/split during this alignment is used to calculate the accuracy.

$$\mathrm{Accuracy} = (1 - (\mathrm{EC}/\mathrm{N})) * 100 \tag{8}$$

Here *EC* is the number of edit operations required to match the system output LCU with human-identified LCUs and *N* is the number of sentences in the document. Our overall accuracy for identifying LCUs was 78.26%. Details are tabulated below in Table 4.

Table 4. Local Coherent Unit Identification Accuracy

DocNo.	Edits	Sentences	Accuracy%
1	10	52	80.76
2	35	134	73.88
3	16	68	76.47
4	14	55	70.91
5	6	56	89.29

4.4 Overall Summary Quality

In order to test the overall readable quality and coherence of our summary we performed a readability evaluation experiment in which 6 participants were given pairs of summary - one generated by state-of-the art summarization system by G-Flow and the other generated by our system - for all the clusters in the DUC 2004 dataset. The 6 evaluators were the research students of Computational Linguistics, who could effectively decide the summary quality in terms of readability and coherence. The two candidates of summary pair were shown in random order and the evaluators had to choose which candidate summary rated better. If the evaluator was ambiguous about his choice he could stay indifferent and mark the rating as 'ambiguous'. As seen below, the preference for our system is more than the G-Flow.

We also have compared the summary quality of the system with and without performing topic segmentation. The overall quality of the summary was higher when topic segmentation is performed.

Table 5. Overall Summary Preference

Our approach	G-Flow	Ambiguous
47%	41%	12%

Table 6. Summary preference within our approach

With topic segmentation	Without topic segmentation	Ambiguous
60%	33%	7%

5 Conclusion

Treating summarization as a content coverage optimization problem by selecting individual sentences as candidates can achieve a flexible content coverage but may result in incoherent summary. We have treated summarization not just as an optimization of content coverage but also have retained the inter-sentence structural relationships at the level of LCU intact. For now, we assumed a linear structure for the local coherent unit(LCU) as a starting point for the approach.

Going forward we can incorporate a graphical structure for a local coherent unit which gives a more noise-free LCUR. We have used a variant of concept coverage algorithm without any corpus dependent features which makes this approach general enough for a domain-independent summarization. The merits of HLDA topic model can be better realized for real-time bigger documents which have better paragraph organization structure thus improving ordering of sentences.

References

1. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1998) 335–336
2. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40** (2004) 919–938
3. Qazvinian, V., Radev, D.R., Özgür, A.: Citation summarization through keyphrase extraction. In: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics (2010) 895–903
4. Shen, C., Li, T.: Multi-document summarization via the minimum dominating set. In: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics (2010) 984–992
5. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research (2004) 457–479
6. Haghighi, A., Vanderwende, L.: Exploring content models for multi-document summarization. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 362–370

7. Celikyilmaz, A., Hakkani-Tur, D.: A hybrid hierarchical model for multi-document summarization. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 815–824
8. Li, P., Wang, Y., Gao, W., Jiang, J.: Generating aspect-oriented multi-document summarization with event-aspect model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011) 1137–1146
9. Aker, A., Cohn, T., Gaizauskas, R.: Multi-document summarization using a* search and discriminative training. In: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics (2010) 482–491
10. Galanis, D., Lampouras, G., Androutsopoulos, I.: Extractive multi-document summarization with integer linear programming and support vector regression. In: COLING, Citeseer (2012) 911–926
11. Woodsend, K., Lapata, M.: Multiple aspect summarization using integer linear programming. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics (2012) 233–243
12. Berg-Kirkpatrick, T., Gillick, D., Klein, D.: Jointly learning to extract and compress. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 481–490
13. Takamura, H., Okumura, M.: Text summarization model based on maximum coverage problem and its variant. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 781–789
14. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 510–520
15. Li, P., Deng, G., Zhu, Q.: Using context inference to improve sentence ordering for multi-document summarization. In: IJCNLP. (2011) 1055–1061
16. McKeown, K., Hatzivassiloglou, V., Barzilay, R., Schiffman, B., Evans, D., Teufel, S.: Columbia multi-document summarization: Approach and evaluation. (2001)
17. Lapata, M.: Probabilistic text structuring: Experiments with sentence ordering. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Association for Computational Linguistics (2003) 545–552
18. Donghong, J., Yu, N.: Sentence ordering based on cluster adjacency in multi-document summarization. IJCNLP 2008 (2008) 745–750
19. Christensen, J., Mausam, S.S., Soderland, S., Etzioni, O.: Towards coherent multi-document summarization. In: HLT-NAACL, Citeseer (2013) 1163–1173
20. McKoon, G., Ratcliff, R.: Inference during reading. Psychological review **99** (1992) 440
21. Marcu, D.: (The theory and practice of discourse parsing and summarization)
22. Foltz, P.W., Kintsch, W., Landauer, T.K.: The measurement of textual coherence with latent semantic analysis. Discourse processes **25** (1998) 285–307
23. Althaus, E., Karamanis, N., Koller, A.: Computing locally coherent discourses. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics (2004) 399
24. Karamanis, N., Poesio, M., Mellish, C., Oberlander, J.: Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics (2004) 391

25. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse **8** (1988) 243–281
26. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A.K., Webber, B.L.: The penn discourse treebank 2.0. In: LREC, Citeseer (2008)
27. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics (2010) 147–156
28. Griffiths, D., Tenenbaum, M.: Hierarchical topic models and the nested chinese restaurant process. Advances in neural information processing systems **16** (2004) 17
29. Kazantseva, A., Szpakowicz, S.: Linear text segmentation using affinity propagation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011) 284–293
30. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. Volume 8. (2004)
31. Nobata, C., Sekine, S.: Crl/nyu summarization system at duc-2004. In: Proceedings of DUC. (2004)

Measuring Diversity of a Domain-Specific Crawl

Pattisapu Nikhil Priyatam, Ajay Dubey, Krish Perumal, Dharmesh Kakadia, and Vasudeva Varma

Search and Information Extraction Lab, IIIT-Hyderabad, India
{nikhil.pattisapu, ajay.dubey, krish.perumal,
dharmesh.kakadia}@research.iiit.ac.in, vv@iiit.ac.in

Abstract. In this work we present various metrics to measure diversity of a domain-specific crawl. We evaluate these metrics using domain-specific crawl originated from ODP URLs and find that these metrics are indeed able to capture diversity. We argue that these metrics can be used for comparing seed sets and crawling strategies with respect to diversity.

1 Introduction

With exponentially increasing content on Internet the use of domain specific search engines are on rise [13]. Several researchers have argued that using focused crawlers for building domain specific search engines is more efficient [5] [11]. Traditionally, a focused crawler is judged based on its ability to fetch relevant (to a domain) documents. While document relevancy is an important aspect of retrieval, crawl diversity of content is equally important factor that impacts the quality of a domain specific search engine. To a large extent, crawl diversity depends on the focused crawler of a search engine. The efficiency of a focused crawler in turn depends on the choice of seed URLs [12] (the list of URLs that the crawler starts with) and the crawling strategy it uses. Though significant effort has gone into building various focused crawlers, not enough research has been done in evaluating them. Even the very few ones which evaluate focused crawlers rely on metrics related to precision, harvest ratio (rate of change of precision), crawl robustness, etc. To the best of our knowledge, there is no work which discusses metrics to measure diversity of a domain-specific crawl.

In this work we propose various metrics to measure diversity of a **domain-specific** web crawl. To achieve this, we use four different methods: semantic distance, statistical dispersion, average similarity and topic modeling. These metrics can be used to better analyze and compare different seed sets and focused crawling strategies as shown in figure 1. In figure 1, we say Seed set S1 is more diverse than Seed set S2, if the diversity score of crawl C1 is strictly greater than that of C2. The same applies for comparing focused crawlers as well.

2 Related Work

The notion of diversity has received great attention in the problem of search results diversification [8][4][18][16][14]. The problem of maximizing search results

pp. 313–321; rec. 2015-01-21; acc. 2015-02-27

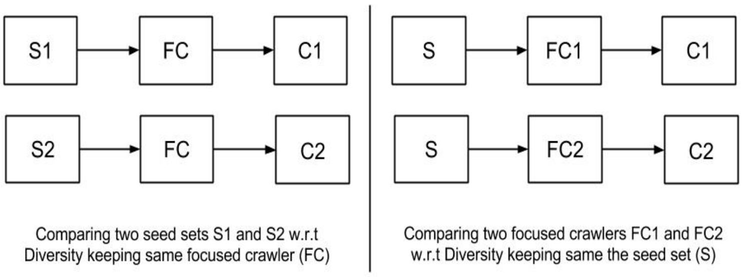

S1
FC
C1
S2
FC
C2
Comparing two seed sets S1 and S2 w.r.t Diversity keeping same focused crawler (FC)
S
FC1
C1
S
FC2
C2
Comparing two focused crawlers FC1 and FC2 w.r.t Diversity keeping same the seed set (S)

Fig. 1. Comparing Seed sets and Focused Crawlers based on crawl diversity

coverage with respect to different aspects of a query is NP-hard [1]. Most previous works on search result diversification are based on a greedy approximation to this problem [17].

Santos et al. use a sub-query based method for search results diversification [17][15]. They assume queries submitted to a retrieval system are ambiguous. Based on this assumption they submit several sub-queries to the retrieval system wherein each sub-query captures a different “aspect” of the query. They present to the user the merged list of ranked retrieved documents.

Dou et al. [3] argue that search results should be diversified in a multi-dimensional way, as queries are usually ambiguous at different levels and dimensions. They mine subtopics from anchor texts, query logs, search result clusters, and web sites and propose techniques to diversify search results based on multiple dimensions of subtopics. They claim that, by incorporating multiple types of subtopics, their models improve the diversity of search results over the sole use of one of them.

Apart from search results, diversity also plays a crucial role in recommender systems. Zhou et al. [19] mention that the key challenge in making useful recommendation is that while the most useful individual recommendations are to be found among diverse niche objects, the most reliably accurate results are obtained by using user or object similarity. They come up with a hybrid approach to resolve this dilemma. They use two features to judge the diversity, which they call ’personalization’ (inter-user diversity) and ’surprisal/novelty’ (capacity of the recommender system to generate novel and unexpected results to suggest objects a user is unlikely to know about already).

As discussed above, there are several metrics to evaluate retrieval systems and recommender systems w.r.t diversity, but there is no single work which evaluates a focused crawler w.r.t diversity. Even the very few ones which evaluate focused crawlers rely on metrics related to precision, harvest ratio (rate of change of precision), crawl robustness, etc. Menczer et al. [7] propose various methods for evaluating topic specific crawl. In the *Assessment via Classifiers* method, they train a classifier for each topic and evaluate the precision of the crawled set. This requires huge amount of accurate training data (manual tagging) which is labour intensive. The second method *assessment via a retrieval system* is based on the intuition that a crawler should retrieve good pages earlier than the bad ones. The last method *Mean Topic Similarity* measures the cohesiveness of the crawled set with the topic as the core. The underlying assumption is that the more cohesive the crawled set the more relevant its pages. To the best of our knowledge, there is no work which discusses metrics to evaluate a domain specific crawl.

3 Approach

In this section we present various approaches to measure the diversity of a given web crawl. Each web page is represented in the form of a text document which contains the parsed text of the web page. In the rest of this paper we refer to a web page as a document.

3.1 Semantic Distance

This measure uses semantic distance of documents within a crawl to calculate its diversity. The main intuition behind this metric is that a more diverse web crawl will have a higher semantic distance between its documents when compared to a less diverse web crawl. Semantic distance between two documents is defined as the average Wordnet distance between their top k keywords (described in equation 1). The crawl diversity(D1 score) is then computed as the average semantic distance between every document pair as shown in equation 2, where N represents the total number of documents present in the crawl. The Wordnet distance is calculated using Wordnet similarity as used by Pedersen et al. [10]. The Wordnet distance function is designed in a way that highly similar words (or same words) get a score 0 and highly dissimilar words get a score of 1.

$$SD(d_x, d_y) = \sum_{i=1}^{k} \sum_{j=1}^{k} Wordnet\ Distance(w_{xi}, w_{y_j})^1 \tag{1}$$

where $SD(d_x, d_y)$ represent the semantic distance between documents x and y respectively and w_{xi}, w_{y_j} represents i^{th} word of document x and j^{th} word of document y respectively.

$$D1\ Score\ = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} SD(d_i, d_j)}{N^2} \tag{2}$$

[1] `http://rednoise.org/rita/wordnet/documentation/riwordnet_method_getdistance.htm`

3.2 Dispersion around a single centroid

Dispersion refers to the spread or variability in a variable. We measure the variance across the crawled set of documents to judge its diversity. The D2 score or variance, as shown in equation 3, is calculated as the average squared distance of all documents from the mean. Here $\boldsymbol{d_i}$ refers to i^{th} document represented as a vector, $\boldsymbol{\mu}$ represents the mean of all $\boldsymbol{d_i}\ 's$ and N represents total number of documents.

$$D2\ Score\ = \frac{\sum_{i=1}^{N}(\boldsymbol{d_i} - \boldsymbol{\mu})^2}{N} \tag{3}$$

In the above equation we have represented each document as a vector. We consider two feature spaces i.e. bag of words and context vectors.

Bag of Words In this model each document is represented as bag-of-words or a vector of words over entire vocabulary. The value corresponding to each word is its frequency within the document.

Context Vectors Since the bag of words model suffers from the problem of sparsity, we also use the context vector model. The context vector is a much more compact representation of a document, where the document is represented as a centroid of the top n word vectors. Figure 2 shows the context vectors of two documents d_A, d_B (shown in red) with word vectors w_{A1}, w_{A2} and w_{B1}, w_{B2} (shown in black) respectively. The word vectors of both the documents are projected onto a common feature space consisting of words occurring in both the documents. Figure 2 depicts the common space as containing the words i and j. A word vector is a vector of k words surrounding it, where k can be understood as the window size. Again, the value corresponding to each word is its frequency within the document. The concept of context vectors is explained in detail in [9].

3.3 Average Similarity between Document Pairs

In this metric, diversity is measured using the average cosine similarity between every pair of documents. This is shown in the crawl shown in equation 4. The crawl diversity is then calculated as shown in equation 5. The intuition behind this metric is that, higher the similarity between the documents, lesser is the diversity of the crawl. We use two document representations of a document in this metric as well. The diversity is calculated as the inverse of average cosine similarity.

$$ACS = \frac{\sum_{i=1}^{k}\ \sum_{j=1,\ j\neq i}^{k} Cosine\ Similarity(d_i, d_j)}{number\ of\ document\ pairs} \tag{4}$$

$$D3\ Score = \frac{1}{ACS} \tag{5}$$

3.4 Using Topic Models

In this metric, we run LDA [2] on n documents randomly picked from the crawl to get k topics. The D4 score is calculated as the sum of KL divergence [6] values between every two topics as shown in equation 6.

$$D4\ Score = \sum_{i=1}^{k} \sum_{j=1,\ j \neq i}^{k} KLDivergence(t_i, t_j) \tag{6}$$

where t_i and t_j represent topic i and topic j, and

$$KLDivergence(t_i, t_j)\ = \sum_{v=1}^{|V|} ln\left(\frac{t_i(v)}{t_j(v)}\right) t_i(v) \tag{7}$$

where $t_i(v)$ and $t_j(v)$ represent the probabilities of word v in topics i and j respectively and $|V|$ represents vocabulary size. Hence a web crawl covering varied topics will have a higher diversity score than the crawl containing similar topics.

4 Evaluation Methodology

In this section we validate the metrics discussed in section 3. We argue that a crawl diversity metric is valid if and only if, it satisfies the constraint: *diversity score of a web crawl, generated by a diverse seed URL set is strictly greater than diversity score of a web crawl, generated by a less diverse seed URL set.* For generating a diverse and a less diverse domain-specific crawl, we selectively pick two sets of URLs of same size from ODP. The first set contains URLs of a wide range of topics of a particular domain and the second set contains URLs from the subset of these topics. For instance, in case of a *Health* specific crawl a set of ODP URLs under the top level categories - *aging*, *fitness*, *nutrition*, *insurance*, etc. would correspond to a diverse set of URLs. ODP URLs under the deeper level category of *dietitian* would correspond to a less diverse set of URLs. This is illustrated in figure 3. We crawl these two sets of URLs thus giving rise to more diverse and less diverse web crawls.

5 Experimental Setup

For this work, we experiment on three domains - tourism, health and sports. For tourism, we pick lodging as the corresponding less diverse seed set. Similarly, for health and sports, we pick dietitian and badminton respectively. 200 URLs are picked from each of these to be used as seed URLs, and a depth 1 crawl is

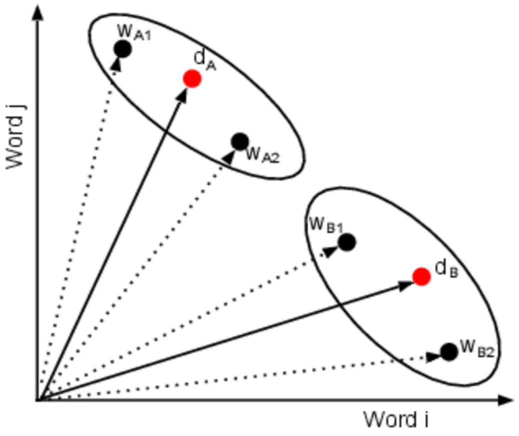


Fig. 2. Context Vectors

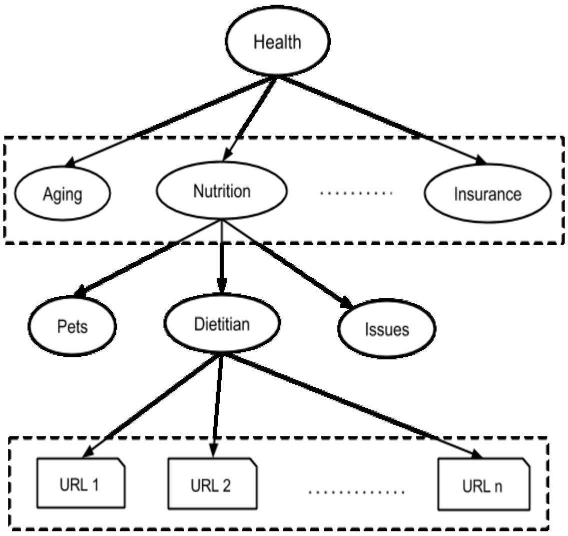


Fig. 3. Picking URLs from ODP Hierarchy

performed. These URLs and their respective crawls have been publicly released for research purposes [2]. In the methods using context vectors, we use 5 context vectors per document and word vectors are generated using a window of 4 surrounding words. In case of the topic modeling approach, we use number of topics as 5. The hyper-parameters for LDA are α is 50 and β is 0.01.

[2] https://www.dropbox.com/sh/i66hzq5cu9aq50k/0_idXZ5s0L

6 Results

Tables 1, 2, 3 and 4 show the diversity scores for the metrics based on semantic distance, topic modeling, average similarity between document pairs and dispersion respectively. MD refers to the more diverse crawl (i.e. crawl originated from diverse seed set) and LD refers to the less diverse crawl (Crawl originated from less diverse seed set). The term "Ratio" refers to the ratio of diversity score of MD to diversity score of LD.

	MD	LD	Ratio
Tourism	0.788	0.779	1.01
Health	0.759	0.721	1.05
Sports	0.764	0.769	0.99

Table 1. Semantic Distance Based Metric

	MD	LD	Ratio
Tourism	4.475	4.470	1.001
Health	4.645	4.406	1.054
Sports	4.470	4.420	1.011

Table 2. Topic Modeling Based Approach

Feature Space	Bag of Words			Context Vectors		
	MD	LD	Ratio	MD	LD	Ratio
Tourism	53.19	31.35	1.69	83.33	46.26	1.77
Health	49.54	18.61	2.66	83.34	26.52	2.91
Sports	37.60	14.07	2.67	55.55	18.21	3.04

Table 3. Similarity Based Metric

Feature Space	Bag of Words			Context Vectors		
	MD	LD	Ratio	MD	LD	Ratio
Tourism	118.31	109.42	1.08	38.37	36.82	1.04
Health	169.58	110.56	1.53	52.79	37.07	1.42
Sports	118.55	100.48	1.17	40.70	36.21	1.12

Table 4. Dispersion Based Metric

7 Analysis and Insights

We observe that the diversity measure based on average similarity between document pairs outperforms the rest of the approaches. We also find that the semantic distance metric fails to distinguish diverse and less diverse crawls. In fact, it wrongly identified a diverse sports crawl as being less diverse and vice versa i.e. $\frac{D(MD_{sports})}{D(LD_{sports})} < 1$. Upon deeper inspection, we find that many words and their spell variations are not present in Wordnet. This has adversely affected the performance of the metric. Also, we find that our crawl contains ill-parsed and non-English language documents because of which meaningful topics were not formed, thereby affecting the performance of the topic modeling based approach.

8 Conclusions and Future Work

This work presents four metrics to measure diversity of a domain-specific crawl which are useful in the context of domain-specific search engines. We rank these metrics based on their ability to differentiate between crawls originated from diverse and less diverse seeds. From the experiments on the tourism, health and sports domains, we observe that the cosine similarity based metric outperforms all others. In both cosine similarity and dispersion based measures context vectors proved to be a better feature space than bag of words. All the proposed metrics, except the semantic distance based metric, are language independent. Even, the semantic distance metric can be easily extended to other languages for which a concept hierarchy like Wordnet is available.

In future, we would like to work on the relevant and correctly parsed portions of the crawl with the help of state of the art parsers and classifiers. The proposed metrics can be used to better analyze and compare different focused crawling strategies. Moreover, more efficient focused crawlers can be built by analyzing the diversity of the previously crawled content, thus leading to higher diversity of the resultant domain-specific crawl. The current work does not use domain knowledge or any external resource to evaluate the crawl diversity. In future, we wish to use the subtopic structure of a domain to evaluate crawl diversity.

References

1. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
3. Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen. Multi-dimensional search result diversification. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 475–484. ACM, 2011.
4. M. Drosou and E. Pitoura. Search result diversification. *ACM SIGMOD Record*, 39(1):41–47, 2010.
5. D. Fesenmaier, H. Werthner, and K. Wober. Domain specific search engines. In *Travel Destination Recommendation Systems: Behavioural Foundations and Applications*, pages 205–211. CABI, 2006.
6. S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
7. F. Menczer, G. Pant, P. Srinivasan, and M. Ruiz. Evaluating topic-driven web crawlers. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 241–249. ACM, 2001.

8. E. Minack, G. Demartini, and W. Nejdl. Current approaches to search result diversification. In *Proc. of 1st Intl. Workshop on Living Web*. Citeseer, 2009.
9. S. Patwardhan and T. Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, volume 1501, pages 1–8, 2006.
10. T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.
11. N. Priyatam, V. Reddy, and V. Varma. Domain specific search in indian languages. In *Proceedings of the First Workshop on Information and Knowledge Management for Developing Regions (IKM4DR) at CIKM 2012*, pages 23–30. ACM, 2012.
12. P. N. Priyatam, A. Dubey, K. Perumal, S. Praneeth, D. Kakadia, and V. Varma. Seed selection for domain-specific search. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 923–928. International World Wide Web Conferences Steering Committee, 2014.
13. P. N. Priyatam, S. Iyengar, K. Perumal, and V. Varma. Don't use a lot when little will do: Genre identification using urls. *Research in Computing Science*, 70:207–218, 2013.
14. F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692. ACM, 2006.
15. R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890. ACM, 2010.
16. R. L. Santos, C. Macdonald, and I. Ounis. Aggregated search result diversification. In *Advances in Information Retrieval Theory*, pages 250–261. Springer, 2011.
17. R. L. Santos, J. Peng, C. Macdonald, and I. Ounis. Explicit search result diversification through sub-queries. In *Advances in information retrieval*, pages 87–99. Springer, 2010.
18. R. H. van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual diversification of image search results. In *Proceedings of the 18th international conference on World wide web*, pages 341–350. ACM, 2009.
19. T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

A study of Impact of Integration Structural Context in Multimedia Retrieval: Application on Image Media

Sana FAKHFAKH, Mohamed TMAR, and Walid MAHDI

Laboratory MIRACL, Institute of Computer Science and Multimedia of Sfax, Sfax University, Tunisia
sanafakhfakh@yahoo.fr,mohamed.tmar@isimsf.rnu.tn,walid.mahdi@isimsf.rnu.tn
http://www.miracl.rnu.tn

Abstract. The expansion in volume of information organized according to a structure in multiple applications introduces a new equivocal in multimedia retrieval in semi-structured documents. We study in this paper the impact of introduce a structural context on multimedia retrieval in XML document thus we present a indexing model which combines textual and structural information. We propose a geometric method who use implicitly of textual and structural context of XML elements and we are particularly interested by improve the effectiveness of various structural factors for multimedia retrieval. Using a geometric metric, we can represent structural information in XML document with a vector for each element. Experimental evaluation is carried out using the INEX Ad Hoc Task 2007, ImageCLEF Wikipedia Retrieval Task 2010 and ImageCLEF Plant Task 2014 in the framework of our participation in CLEF 2014 campaign. The results show that integration of structural context significantly improves compared results of using a single textual context. Our proposed method perform as compared to other systems evaluated on two coolection INEX 2007 and ImageCLEF 2010.

Keywords: Structural context, Textual context, Approximative resolution, XML element, Image retrieval

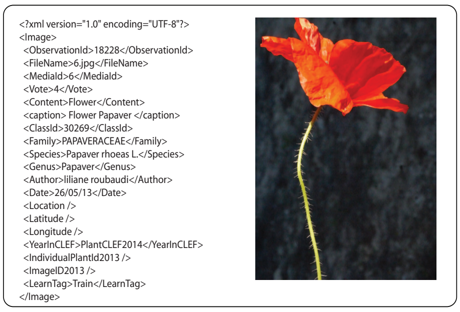
1 Introduction

The joint evolution of user needs and electronic semi-structured documents constantly raises new challenges in the Information Retrieval (IR) field. The need with this kind of information is justified by quick change of scopes of application which use structural documents (format HTML or XML) what imposes new challenges in the field of search for information. Indeed, nowadays XML document passed a simple tool for exchanging data to a new storage medium. XML document includes textual element and multimedia element such as image, audio and video. These elements are organized according to structure which includes information notably although there is not only one manner to organize contents. However, the choice of structure depends greatly on the context of use of the textual contents. Mainly in the literature, there are two main classes of approaches in the field of multimedia retrieval: retrieval methods based on multimedia content (MR-content) and multimedia methods to retrieval based on context (MR-Context).

The approaches of the multimedia retrieval based on content use specific features of low level according to type of media [1]. We can cite for example image retrieval that exploits visual features (the color, texture, forms $\cdots$). These methods have proven effective with media "image" in well defined fields such as medical field this is due to requirement for thorough knowledge of distinctive media. This type of research can be applied to only one type of media in system due to lack of semantic representation in media content.

The approaches of the multimedia retrieval based on context do not depend on type of media in question [2] [3]. Indeed, these methods rely on information surrounding the multimedia element representing its semantic description. Multimedia retrieval based on textual context is most used, although the structural context remains an obvious source which plays a part paramount in understanding of structured documents.

Multimedia retrieval based on textual context is most used, although the structural context remains an obvious source which plays a part paramount in understanding of structured documents. In this paper, we are interested in Context-based MIR techniques, and more precisely in MIR based on textual and structural context in XML documents. Image context is composed all textual information surrounding the image. For retrieve image presentated in Figure 1, we can use text surrounding image such as document title, image name, image caption, etc



```
<?xml version="1.0" encoding="UTF-8"?>
<Image>
 <ObservationId>18228</ObservationId>
 <FileName>6.jpg</FileName>
 <MediaId>6</MediaId>
 <Vote>4</Vote>
 <Content>Flower</Content>
 <caption> Flower Papaver </caption>
 <ClassId>30269</ClassId>
 <Family>PAPAVERACEAE</Family>
 <Species>Papaver rhoeas L.</Species>
 <Genus>Papaver</Genus>
 <Author>liliane roubaudi</Author>
 <Date>26/05/13</Date>
 <Location />
 <Latitude />
 <Longitude />
 <YearInCLEF>PlantCLEF2014</YearInCLEF>
 <IndividualPlantId2013 />
 <ImageID2013 />
 <LearnTag>Train</LearnTag>
</Image>
```

Fig. 1. Example of a multimedia element context.

The textual context remains insufficient in most of time. In this context, [4] say: "Ignore the document structure is to ignore its semantics". There are other sources of evidence that were used as visual descriptors, information from link around the image, structure of XML document. Indeed, We focus on XML documents don't have a homogeneous structure. What makes the structure as new source of evidence.

In this article, we focus on techniques for multimedia retrieval based on textual and structural context in XML documents. This type of document includes textual information and structural constraints. So, XML document cannot be effectively exploited by classical techniques of IR, which regard document as a plane source of information. The implicit incorporation of multimedia elements in XML documents requires the exploitation of textual context for multimedia retrieval. However, the textual context remains insufficient in most of time. The idea is to calculate the relevancy score of media element based on information from the textual and structural context to answer a specific information needs of user, expressed as query composed of set of keywords.And seeking the most appropriate manner to combine two sources of evidence: text and structure. Our main inspiration is to use the structure to involve each textual information depending on its position in XML document, that is textual information that gives the best possible description of multimedia element. In our work, we will be interested by media "image". Most existing work in this area uses the information from textual description of image. There are other sources of evidence that were used as visual descriptors, information from link around the image [5], structure of XML document. To resolve difficulties in mutlimedia retrieval field, you must define adequate source of evidence for representation a multimedia element and defining appropriate indexing model. In this context, we present our structural indexing system combining conceptual information for semi-structured documents dedicated to approximate retrieval data. In Section 3, we presents the details of our proposed method where detailing the preprocessing, extraction of textual and structural and phase calculation relevance of multimedia element in information a better response to needs expressed by user and we describe our structural indexing system combining conceptual information for semi-structured documents dedicated to approximate retrieval data. In Section 5, we presents the results of her applying on three data sets "INEX 2007", "ImageCLEF 2010" and "ImageCLEF 2014". The last section provides our conclusions and future works.

2 RELATED WORKS

The advent of structured documents has caused new problems in information retrieval world, and more specifically in multimedia elements retrieval. These problems are strongly related to nature of these documents that provide the structure as a new source of evidence. Thus, nowadays, XML documents include multimedia elements of different types (audio, video and image)implicitly embedded in the textual elements. These multimedia elements (such as physical objects) do not contain enough information to be able to answer a given query. Therefore, the calculation of relevance score of multimedia element must be linked to textual and structural information provided by other nodes XML [5]. Indeed, XML document is used to describe a set of data by a structure that provides a semantic lexicon. Thus, it facilitates the presentation of information in terms of interpretation and exploitation. Replying to this need, new works appear in the field of multimedia retrieval that takes in account the structure as source of relevant information. Existing work in structured retrieval of multimedia elements is decomposed in two classes.

The first class includes some works which proceed to adopt some traditional technical of retrieval information as language model. In this context, the team *CWI/UTwente* performs a step of filtering results to keep the fragments containing at least one multimedia element [6][7].

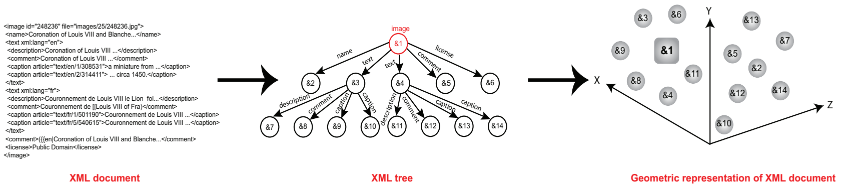
The second class includes the specific work to be structured multimedia retrieval. This class uses the structure as a source of evidence in the process of selection of multimedia elements. As first step, [8] proposed a method which combines structure of XML document (XPath) with the use of links (XLink). This method consist to divide XML document into regions. Each region represent a area of ancestors of the multimedia element. His score is calculated in function of the scores of each region. This method exploits vertical structure only. In a second time, [9] have used the addition of horizontal structure to the notion of hierarchy. [9] use a method called "CBA" (Children, Brothers, Ancestors), which takes into consideration the information carried by the children , brothers and fathers nodes for calculate the relevance of multimedia elements. The authors propose an alternative method "OntologyLike" which is based on the identification of XML document to ontology. To calculate the similarity between nodes the authors use similarity measures that are mainly based on the number of edges to calculate the distance between nodes.

There are other approaches to multimedia retrieval are based on exploitation of links in XML document [10]. This work was improved by proposing a hybrid approach that combines structure with using of links who is consider as semantic links [11]. This method above to divide the document into regions according the hierarchical structure and the location of image in document. This factor plays a role in the weighting of links for compute the score of image.

In this paper, we propose a new metric for multimedia retrieval in XML documents which involves the use of geometric distances to calculate the relevance of each node from the multimedia node. This method consists of placing the nodes of XML document in Euclidean space and define each node by a vector of coordinates to calculate then the distance between each pair of nodes. This distance will play a beneficial role in to calculate the score of multimedia element.

3 Proposed Approach

We focus on techniques for multimedia retrieval based on textual and structural context in XML documents. XML documents cannot be effectively exploited by classical techniques of IR, which regard document as a bog of words. Therefore, the calculation of relevance score of multimedia element must be linked to textual and structural information provided by other nodes XML [5]. Thus, it facilitates the presentation of information in terms of interpretation and exploitation. Replying to this need, we propose a new method in the field of multimedia retrieval that takes into account the structure as a source of evidence and its impact on search performance. We present a new source of evidence dedicated to multimedia retrieval based on the intuition that each textual node contains information that describes semantically a multimedia element. And the participation of each text node in the score of a multimedia element varies with its position in there XML document. To compute the geometric distance, we initially place the nodes of each XML document in an Euclidean space to calculate the coordinates of each node by algorithm 1. Then, we compute the score of a multimedia element depending on the distance between each textual node. Figure 2 shows the steps of passing an XML document to a geometric representation of the XML elements in a Euclidean space. The first step consist to present a XML document as XML tree to take into account XML document properties. An XML tree is described by a set of relationships between nodes. Formally an XML tree is a pair $A = (E, R)$ where E is a set of XML elements and $R \subset E^2$, $((p, q) \in \ R$ if p is the parent of $q)$ is a set of relations satisfying:



```
<image id="248236" file="images/25/248236.jpg">
 <name>Coronation of Louis VIII and Blanche...</name>
 <text xml:lang="en">
  <description>Coronation of Louis VIII ...</description>
  <comment>Coronation of Louis VIII ...</comment>
  <caption article="text/en/1/308531">a miniature from ...</caption>
  <caption article="text/en/2/314411"> ... circa 1450.</caption>
 </text>
 <text xml:lang="fr">
  <description>Couronnement de Louis VIII le Lion  fol...</description>
  <comment>Couronnement de [[Louis VIII of Fra)</comment>
  <caption article="text/fr/1/501190">Couronnement de Louis VIII ...</caption>
  <caption article="text/fr/5/540615">Couronnement de Louis VIII ...</caption>
 </text>
 <comment>({{en|Coronation of Louis VIII and Blanche...</comment>
 <license>Public Domain</license>
</image>
```

Fig. 2. The steps of passing an XML document to geometric representation.

$$\exists! r \in E, \forall q \in E - \{r\}, (r, q) \in R \tag{1}$$

With r is the root of the tree.

$$\forall p \in E - \{r\}, \exists! q \in E, (p, q) \in R \tag{2}$$

Each node has a parent except the root r. In second step, we will spend to presentation of XML tree in a geometric representation. This step is mainly based on equalities extraction in XML tree according to our proposed hypotheses. The XML tree representation allowed us to unveil certain relationships of neighboring, brotherhood and offspring. Indeed, the distance d which separate two or more brothers with their common ancestors iteratively is the same. And brothers of the same hierarchical level are equidistant. These distances are defined according to the relationship of contiguity and semantic similarity between nodes. These distances are not quantized but will be extracted in function of the position of each textual node in XML tree. All these properties result in: For all $q_i = (x_{i1}, x_{i2} \cdots x_{im})$ and $q_j = (x_{j1}, x_{j2} \cdots x_{jm})$ where Q is a set of vectors in $\mathbb{R}^m$.

- In the same hierarchy, if there are more than two brothers then their adjacent nodes are equidistant:
  property 1

  $$\forall q_i, q_j, q_k \in Q,\ if\ A_1(q_i) = A_1(q_j) = A_1(q_k)$$
  $$d(q_i, q_j) = d(q_i, q_k)$$

- The distance between any node and its descendants is the same:
  property 2

  $$\forall q_i, q_j, q \in Q, n \in \mathbb{N}, A_n(q_i) = A_n(q_j) = q$$
  $$d(q_i, q) = d(q_j, q)$$

With $\forall n \in \mathbb{N}^*$, we define function A_n by: $\forall q \in E$,

$$A_n(q) = \begin{cases} \{q\} \text{ if } n = 0 \\ A_{n-1}(p) \text{ if } \exists\ p \in\ E,\ (p,q) \in\ R \text{ and } n > 0 \\ \varnothing \text{ else} \end{cases}$$

From these relationships, we can generate system of equations taking into account for kinship relationships nodes based on hierarchy and adjacency. These relationships are decried by equalities in this order (these equations are only examples)(Figure 2):

$$\begin{array}{c} d(n_1, n_2) = d(n_1, n_3) \\ d(n_1, n_2) = d(n_1, n_4) \\ d(n_1, n_7) = d(n_1, n_8) \\ d(n_1, n_7) = d(n_1, n_9) \end{array}$$

These distances are defined according to the relationship of contiguity and semantic similarity between nodes. They are not quantized but will be extracted in function of the position of each textual node in the XML tree. The resulting system is nonlinear, its resolution requires the use of an approximate resolution iteratively method where we used iterative solution method (see Algorithm 1). The process begins by assigning to each XML node a random vector. It Tries to improve the coordinate values of each node according to an error value (the sum of the squared deviations). At each iteration, the coordinates are improved together with the minimization of this error. The algorithm stops when the error reaches its minimum value (no improvement is possible). Let Q the set of vectors obtained at a given iteration during the running of the algorithm, the error is defined by:

$$\begin{aligned} error(Q) = & \sum_{\substack{q_i, q_j, q_k \in Q, \\ A_1(q_i) = A_1(q_j) = A_1(q_k)}} (d(q_i, q_j) - d(q_i, q_k))^2 \\ & + \sum_{\substack{q_i, q_j, q \in Q, n \in \mathbb{N}, \\ A_n(q_i) = A_n(q_j) = q}} (d(q_i, q) - d(q_j, q))^2 \end{aligned}$$

Algorithm 1 Resolution algorithm approximate nonlinear system of equations

```
Require: $(Q = (q_1, q_2 ... q_{|Q|}), R)$ :an XML tree as $q_i$=$(q_{i1}$,$q_{i2}$...$q_{im})$  $\forall i \in [1, |Q|]$
  $m$:dimension
  for $(i, j) \in [1, |Q|]^2$ do
    $q_{ij} \leftarrow$ random value
  end for
  $Q_1 \leftarrow (q_1, q_2 ... q_{|Q|})$
  repeat
    $P \leftarrow Q_1$
    for $(i, j) \in [1, |Q|]^2$ do
      $Q_2 \leftarrow (q_1, q_2 ... q_{i-1}, q_i + d_j(1), q_{i+1} \cdots q_{|Q|})$
      $Q_3 \leftarrow (q_1, q_2 ... q_{i-1}, q_i + d_j(\varepsilon), q_{i+1} \cdots q_{|Q|})$
      $Q_4 \leftarrow (q_1, q_2 ... q_{i-1}, q_i + d_j(1 - \varepsilon), q_{i+1} \cdots q_{|Q|})$
      $t \leftarrow 0$
      while $error(Q_1) > error(Q_2) > error(Q_3) > error(Q_4)$ do
        $Q_4 = (q_1, q_2 ... q_{i-1}, q_i + 2^t d_j(1), q_{i+1} \cdots q_{|Q|})$
        t=t+1
      end while
      $t \leftarrow 0$
      while $error(Q_1) < error(Q_2) < error(Q_3) < error(Q_4)$ do
        $Q_1 = (q_1, q_2 ... q_{i-1}, q_i - 2^t d_j(1), q_{i+1} \cdots q_{|Q|})$
        t=t+1
      end while
      while $|error(Q_1) - error(Q_2)| > \varepsilon$ do
        $Q_5 \leftarrow \frac{Q_1 + Q_2}{2}$
        let $Q_5 = (p_1, p_2 ... p_{|Q|})$
        if $error(p_1, p_2 ... p_{i-1}, p_i - d_j(\varepsilon), p_{i+1} \cdots p_{|Q|}) > error(p_1, p_2 ... p_{i-1}, p_i + d_j(\varepsilon), p_{i+1} \cdots p_{|Q|})$ then
          $Q_1 \leftarrow Q_5$
        else
          $Q_2 \leftarrow Q_5$
        end if
      end while
    end for
  until $P = Q_1$
```

Where m is the dimension of the Euclidean space and $\forall v \in \mathbb{R}$, $D_j(v) = (d_1, d_2 \cdots d_m)$ is such as:

$$d_k = \{ \begin{matrix} 0 \; if \; k \neq j \\ v \; otherwise \end{matrix}$$

3.1 Multimedia Element Representation by Textual and Structural Context

A multimedia element (eg *image*) does not contain textual content. Its score is based on textual nodes in its neighborhood. The transition from the XML tree structure representation of elements in an Euclidean space, where we exploit the dissimilarity distances separating a multimedia node and other textual nodes, is performed by extracting the equations satisfying the properties defined earlier and the application of algorithm 1. To calculate the distance between a node n and multimedia element H, we will try to use several geometric distances such as Manhattan distance, Euclidean distance and Minkowski distance between their respective feature vectors q_n and q_H described by the following equations:

$$dist_{Manhattan}(n,H) = \sum_{i=1}^{m} \mid q_n - q_H \mid \tag{3}$$

$$dist_{Euclidean}(n,H) = \sqrt{\sum_{i=1}^{m} (q_n - q_H)^2} \tag{4}$$

$$dist_{Minkowski}(n,H) = \sqrt[p]{\sum_{i=1}^{m} \mid q_n - q_H \mid^p} \tag{5}$$

With m is the dimension of the Euclidean space and p=1. q_n is defined by: q_n=(xn_{i1}, xn_{i2} ... xn_{im}) with xn are the vector characteristics of node n. And q_H is defined by: q_H =(xH_{i1}, xH_{i2} ... xH_{im} with xH represent the coordinates compose the vector characteristics of a node H. We calculate the score for each textual node depending on the frequency of each term (tf) and the number of elements in the corpus according to the number of elements containing the term (idf). A textual node is presented by: $n = (n_1, n_2 \cdots n_{|v|})$ where n_i is the weight of the term t_i, v is the set of indexing terms:

$$n_i = tf(t_i, n) \times idf(t_i) \tag{6}$$

With

$$idf(t_i) = log(\frac{N}{N_i}) \tag{7}$$

Where N is the total number of XML elements in the corpus, N_i is the number of elements that contain the term t_i and $tf(t_i, n)$ is the frequency of the term t_i in node n. The score of textual node depends on the weight of each indexing term. A query is made by the list $v = (v_1, v_2 \cdots v_{|v|})$ where $v_i \in \{0, 1\}$ (0:not exist, 1:exist) according membership t_i at the query. The score of textual node n for the query q is defined by:

$$rsv(q,n) = q \times n^T = \sum_{i=1}^{|V|} q_i \times n_i \tag{8}$$

Where μ is the set of textual elements. The score of multimedia node H is defined by:

$$rsv(q,H) = \sum_{n \in \mu} \frac{rsv(q,n)}{dist(n,H)} \tag{9}$$

With $dist(n,H)$ is the distance (Manhattan distance or Euclidean distance or Minkowski distance) between the feature vectors corresponding to the nodes n and H. This equation leads to assign the importance of contribution of all nodes in computing the score of multimedia element that shows its beneficial impact in multimedia retrieval.

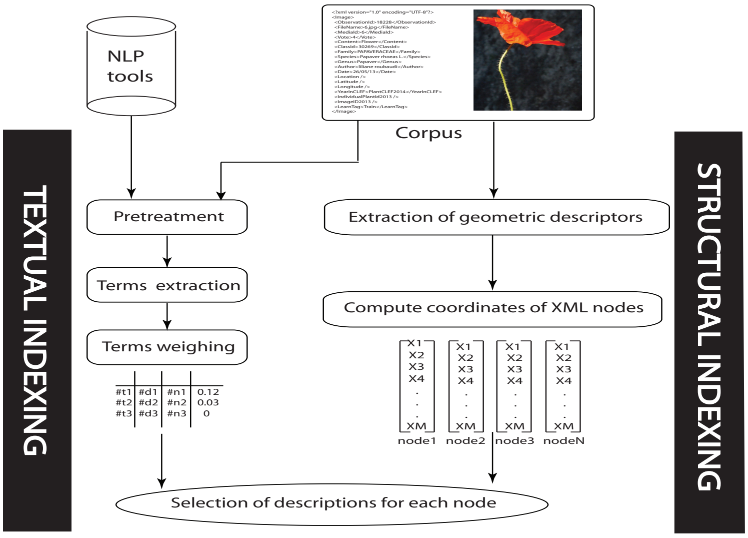
4 Indexing System

We propose a indexing system $MXS - index$ composed by two parts: part of textual indexing and part of structural indexing. Our indexing methodology as schematized in Figure 3. The first part consists of four main steps: Pretreatment, term extraction and term weighing using NLP (Natural Language Processing) techniques to extract the candidate XML nodes of the resulting indexing. The first step is to split text into a set of sentences, prune the stop words for each XML node of the corpus and radicalize terms using the algorithm PORTER [12]. The second step is term extraction and the last step is calculating term importance. That is a fundamental step in information retrieval process and it is determined through term frequency (tf) and inverse document frequency(idf). In Second part, we built structural index using information extract from XML tree and geometric metric. Each XML node will presented by characteristic vector. We start by extract geometric proprieties. And we compute coordinates of each XML nodes. This part is accompanied by generating XML data model which processes ancestor, descendant and proximity relationships (Figure 3). The step of selection of descriptors of each node consists in associating each XML node own these textural and structural descriptors to better combine.

5 Evaluation and Results

We evaluate our system on three databases extracted from three collections : INEX 2007 (Initiative for the Evaluation of XML Retrieval) Ad Hoc task [13], ImageCLEF 2010 Wikipedia image retrieval task [14] and ImageCLEF 2014 Plant task [15]. The first two databases are composed by XML documents extracted from Wikipedia. The latest dataset is collected by scientific community for testing and validation of their approaches (Table 1).

The aim of the experiments in this section is to show the effectiveness of XML structure in multimedia retrieval. For this purpose, we evaluated separately the use of textual context only (**TC**), as well as the combination of the



```
<?xml version="1.0" encoding="UTF-8"?>
<Image>
 <ObservationId>18228</ObservationId>
 <FileName>6.jpg</FileName>
 <MediaId>6</MediaId>
 <Vote>4</Vote>
 <Content>Flower</Content>
 <ClassId>30269</ClassId>
 <Family>PAPAVERACEAE</Family>
 <Species>Papaver rhoeas L.</Species>
 <Genus>Papaver</Genus>
 <Author>liliane roubaudi</Author>
 <Date>26/05/13</Date>
 <Location />
 <Latitude />
 <Longitude />
 <YearInCLEF>PlantCLEF2014</YearInCLEF>
 <IndividualPlantId2013 />
 <ImageID2013 />
 <LearnTag>Train</LearnTag>
</Image>
```

Fig. 3. Architecture of our indexing model $MXS - index$.

Table 1. INEX 2007, ImageCLEF 2010 and ImageCLEF 2014 collections

Company	**INEX 2007**	**CLEF 2010**	**CLEF 2014**
Task	Collection XML Ad Hoc	Wikipedia Retrieval	Plant Retrieval
Number of XML document	659388	237434	47815
Number of image	246730	237434	47815
Topics	19	70	8163

two (**TC and TS**). For INEX 2007, ImageCLEF 2010 and ImageCLEF 2014 test set, we respectively obtain the following MAP values: 0.2376, 0.1674 and 0,2488 using textual context only (figure 4). We compare between the use of Manhattan distance, Euclidean distance and Minkowski distance. We observed that the difference of results between the three distances is very signicant in the INEX 2007 test set, ImageCLEF 2010 test set and ImageCLEF 2014 test set. The Euclidean distance gets a most suitable representation of multimedia element which is none other than the dissimilarity distance between XML nodes. Indeed, the evaluation results show that this distance provides a MAP which is equal to 0.2572 as MAP with using ”ImageCLEF 2010” collection. The result has been improved significantly with the ”INEX 2007” collection to 0.3102 as MAP relative to Manhattan distance (0.2376 for ”INEX 2007” collection, 0.1754 for ”ImageCLEF 2010” collection and 0.2276 for ”ImageCLEF 2014” collection) and Minkowski distance (0.2876 for ”INEX 2007” collection, 0.2245 for ”ImageCLEF 2010” collection and 0.3267 for ”ImageCLEF 2014” collection). This increase is due to nature of ”INEX 2007” collection who includes XML documents with heterogeneous structure. So in ”INEX 2007” collection we find documents with

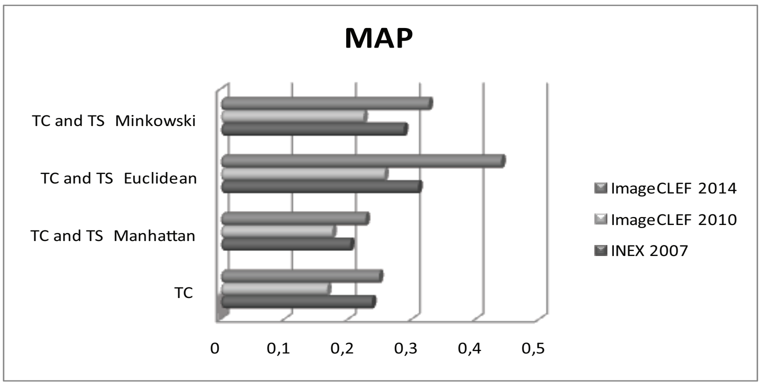


Fig. 4. Results of the impact our approach on INEX 2007, ImageCLEF 2010 and ImageCLEF 2014 based in MAP(Mean Average Precision).

high depth. This factor highlights structural information and amplifies effect textual information based on computed distances. For against, our system is more stable with "ImageCLEF 2010" collection, this is due to rapid convergence of results. With our measure, we have shown that combined use of textual and structural context can properly determine the relevance of multimedia element, and the structure plays a primordial role in multimedia retrieval (Figure4). We can conclude that an using structural information returns better pertinence in case uses multimedia retrieval with using "ImageCLEF 2014" collection with MAP equal to 0.4406. In fact, this collection contains quite specific documents include descriptions of plants which reduces textual concepts; see Figure 4.

After evaluation of our system on the three described collections, we will try to position itself compared to participants in INEX 2007 and ImageCLEF 2010. Our system gives better results using INEX 2007 collection (figure5). Indeed, comparing work proposed by IRIT system, we obtained a MAP equal to 0.31 [13][9]. Our system also gives a better MAP which is equal to 0.25 compared to XRCE system with using textual context with ImageCLEF 2010 collection (figure 6) and he got the better result with textual and visual context a MAP value which is equal to 0.27 [14].

6 CONCLUSION

This approach allowed us to calculate the score of element multimedia according the textual context provided by nodes in proximity and structural context from distance between nodes and multimedia element. This method was evaluated with using of three collections "INEX 2007", "ImageCLEF 2010" and "ImageCLEF 2014". In this work, we studied the impact of textual and structural context on multimedia element retrieval, where the user need can be a multimedia element (text). We plan to investigate the impact of a mixture of text and multimedia element(text+image) with to using visual descriptors.. In

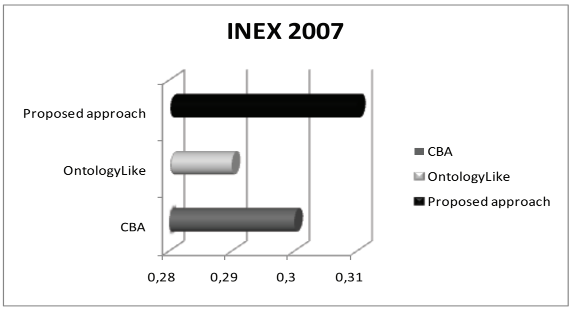


Fig. 5. Compared results on INEX 2007 collection with MAP(Mean Average Precision) metric.

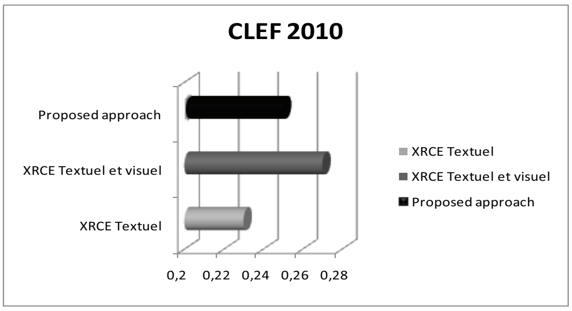


Fig. 6. Compared results on ImageCLEF 2010 collection with MAP(Mean Average Precision) metric.

the future, we want to exploit another factor to calculate the relevance of multimedia element such as the title of image, the weighting of the links in XML document ... As well as another source of evidence as visual descriptors.

References

1. Michael S. Lew. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl*, 2:1–19, 2006.
2. Haytham Elghazel, Khalid Idrissi, Atilla Baskurt, and Chokri Ben Amar. Approche textuelle pour la recherche d'image. In *3rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications SETIT 2005*, March 2005.
3. Dian Tjondronegoro, Jinglan Zhang, Jinfeng Gu, Anthony Nguyen, and Shlomo Geva. Integrating text retrieval and image retrieval in xml document searching. In *INEX*, pages 511–524, 2005.
4. T. Schlieder and M. Holger. Querying and ranking xml documents. *Journal of the American Society for Information Science and Technology*, 53:489–503, 2002.
5. Angelos Hliaoutakis, Giannis Varelas, Epimeneidis Voutsakis, Euripides G. M. Petrakis, and Evangelos Milios. Information retrieval by semantic similarity. In *Intern. Journal on Semantic Web and Information Systems (IJSWIS).Special Issue of Multimedia Semantics*, pages 55–73, 2006.

6. Theodora Tsikrika, Pavel Serdyukov, Henning Rode, Thijs Westerveld, Robin Aly, Djoerd Hiemstra, and Arjen P. Vries de. Structured document retrieval, multimedia retrieval, and entity ranking using pf/tijah. In *6th Initiative on the Evaluation of XML Retrieval, INEX 2007*, volume 4862 of *Lecture Notes in Computer Science*, pages 306–320, London, March 2008. Springer Verlag.
7. Thijs Westerveld, Henning Rode, Roel Os van, Djoerd Hiemstra, Georgina Ramirez, Vojkan Mihajlovic, and A.P. Vries de. Evaluating structured information retrieval and multimedia retrieval using pf/tijah. In N. Fuhr, M. Lalmas, and A. Trotman, editors, *Comparative Evaluation of XML Information Retrieval Systems*, volume 4518 of *Lecture Notes in Computer Science*, pages 104–114, Berlin, Germany, 2007. Springer Verlag.
8. Zhigang Kong and Mounia Lalmas. Xml multimedia retrieval. In *SPIRE*, pages 218–223, 2005.
9. Mouna Torjmen, Karen Pinel-Sauvagnat, and Mohand Boughanem. Using textual and structural context for searching multimedia elements. *IJBIDM*, 5(4):323–352, 2010.
10. Hatem Awadi and Mouna Torjmen. Exploitation des liens pour la recherche d'images dans des documents xml. March 18-20 2010.
11. Hatem Aouadi, Mouna Torjmen-Khemakhem, and Maher Ben Jemaa. Combination of document structure and links for multimedia object retrieval. *Journal of Information Science*, 38(5):442–458, October 2012.
12. M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
13. Norbert Fuhr, Jaap Kamps, Mounia Lalmas, Saadia Malik, and Andrew Trotman. Overview of the inex 2007 ad hoc track. In *INEX*, pages 1–23, 2007.
14. Adrian Popescu, Theodora Tsikrika, and Jana Kludas. Overview of the wikipedia retrieval task at imageclef 2010. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
15. Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors. *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, volume 1180 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

Uyghur Short Text Classification Using Morphological Information

Batuer Aisha

College of Information Science and Engineering, Xinjiang University, Urumqi, Xinjiang, 830046, China

batur61@163.com

Abstract. In this paper, we propose a novel method for improving the classification performance of short text strings using conditional random fields (CRFs) that combine morphological information. Experimental results on three datasets (Uyghur, Chinese, and English) demonstrate that our method can yield higher classification accuracy than Support Vector Machine (SVM) classifier and Maximum Entropy Model (MEM) classifier. Moreover, we show that our method can greatly decrease error rates, particularly if the number of training texts or the size of the strings in the train set is small.

Key words: Uyghur, Short text, Morphological information

1 Introduction

With the rapid growth of the available information on the Internet, text classification is becoming one of the key techniques in organizing, filtering and handling a large amount of text data. Since constructing text classifiers by hand is difficult and time consuming, it is desirable to learn classifiers from training data that do the category assignments automatically. Text classification task has provoked much interest in machine learning. As an aspect of text classification, short text classification is a particular challenge.

Uyghur is an agglutinative language with a rich and complex morphology, which is very different from the morphology of other languages, such as Chinese (Isolating language) and English (Inflecting language). Uyghur is written from right to left and words are separated by a blank space. Observe the following examples:

Sen Kitabxanigha Mang (You go to bookstore) سەن كىتابخانىغا ماڭ

The first and the third words (counted from right to left) are kept unchanged whereas the second word كىتابخانىغا (to bookstore) is decomposed into two parts, غا (inflectional suffix) and, كىتابخانى (corresponding to bookstore), a "quasi-word" that can never appear in the lexicon. After phonetic harmonization [1], the sentence becomes: سەن كىتابخانا غا ماڭ . Note that كىتابخانى is now changed to كىتابخانا (bookstore) which is a word in a lexicon.

It is worth noting that “to bookstore” is a word, or a phrase-like unit in Uyghur, “to” within it is a case-marker-like inflectional suffix. This means that the morphological information of Uyghur words in a sentence is associated with the classification features.

This classification problem is usually viewed as supervised learning, where the goal is to assign predefined category labels to unlabeled documents based on the likelihood inferred from the training set of labelled documents. The text categorization algorithms are to assign each test document set to one or more pre-specified classes [2].

Uyghur information society is facing the challenge of handling massive volume of online documents, news, and so on. The amount of data and increment rate is so high that this process cannot be done by hand. Hence, erotic recognition, filtering of spam mails, monitoring ill gossips and evil messages, quick search of interesting topics from large databases, and retrieving the information based on user’s preferences from information sources, are some other examples where text classification can play an important role. If an automatic classification engine is developed, classification task can be achieved with less cost and in less time, while improving analyst’s productivity. To use the Internet more efficiently, it needs to be classified. When we classify, seek to group things that have a common structure or exhibit a common behavior.

On the contrary of other languages, there is not much study on Uyghur texts. In this study, a system is mainly developed for automatic short text classification of Uyghur texts. The articles are classified into 7 different classes and 68% success ratio is achieved.

As there are no previous researches devoted to short text classification and being short of related works that we could use as a base for our research; in this paper, we compare our CRFs-based approach [3] with Support Vector Machine (SVM) [4] classifier and Maximum Entropy Model[1] (MEM) [5] classifier.

Text classification task has drawn a large body of research in machine learning community [6][7].

As a special aspect of text classification, short text classification is a particular challenge in that, short text examples tend to share few terms. It is particularly difficult to classify new instances and common comparisons between texts because they often yield no useful results. Short texts are typically sparse and ambiguous [8].

Nevertheless, to classify these short texts into certain target categories is a difficult but important problem. It is crucial in many information systems to classify short text segments, such as titles of documents and queries from users, into a well-formed topic hierarchy [9].

For example, in some time-consuming task and because of resource limitations, we deal with the title or the keywords of the document instead of the full text.

On the other hand, many search engines provide users with the ability to use natural language queries to ask questions and search for manually prepared answers. By using query classifications, queries in similar topics can automatically be clustered. Therefore, short text classification can make the preparation of answers to the queries more efficient.

[1] http://homepages.inf.ed.ac.uk/lzhang10/maxent.html

Although widely used, classification of short text snippets, such as titles and search queries, work poorly. Traditional document classification measures are often few, in any terms common between two short text snippets.

Short lengths text segments do not provide enough word or shared context for a good similarity measure. That is the reason why normal machine learning classification approaches usually fail to achieve desired accuracy.

Previous work attempts to overcome the data sparseness to get a better classification performance and deal with the feature sparse problem, external resources are used in short text classification task.

In [10, 11, 12, 13], search engines are employed in order to expand and enrich the context of data. In [14, 15, 16], online data repositories, such as Wikipedia or Open Directory Project, are used as external knowledge sources. With the help of external resource, we can improve the performance. However, the use of external resource, such as the search engine, is quite time-consuming, not suitable for real-time applications.

In order to speed up our short text classification approach, external resources are not used. Instead, we take advantage of the characteristics of short texts to improve the classification algorithm. The underlying idea is that features have strong relationship in very short texts.

The contributions of this paper are twofold. First, we investigate the change of words under different lengths of texts; second, we propose a novel CRFs-based method to improve the short text classification performance.

The remainder of the paper is organized as follows. Section 2 provides our analysis about short text. Based on our analysis, a CRFs-based short text classification method is proposed in section 3. In Section 4, our experimental results are demonstrated. There are different kinds of short text classification applications, so in our experiment section we conduct experiments on different applications to get a comprehensive evaluation. The last section summarizes our contribution and outlines future goals.

2 Consistency of features in very short texts

In this section, we want to investigate the characteristics of short texts. The adjacent features of short texts tend to talk about the same topic. On the other hand, the topic might be changed along with the increase of text length. We will analyze these issues as follows.

2.1 Analysis of short texts

Text segments do not provide enough features in common in short length, so they suffer from the feature sparse problem.

Nonetheless, short texts have their advantage for classification task. Many short texts focus on one topic because of their short lengths. But manual labelling text may be too short to learn new features, and too short to obtain proper word statistics.

Instead, long texts usually contain segments that belong to different topics. For example, a full text about sports may have a segment talking about football, and another segment talking about history. In short texts, this kind of problem seldom happens, because short texts focus on one topic. The features of short texts have strong relationship. If a feature is related to a topic, the features in the near contexts also incline to this topic. In this work, we use tokenization to indicate this kind of characteristic.

In [17], the author proposes a text categorization method in which documents are split into fragments. And instead of classifying the full text, they classify the segments and use the segment category to yield the result. This method can yield some improvements of text classification performance. This work also gives us the clue that text segments have special characteristics. If properly used, it can help us to improve classification accuracy.

3 CRFs-based Classifier

Based on our observation in the previous section that short texts focus on one topic, and that the features of short texts have strong relationship; and that when a feature is related to a topic, the features in the near contexts also incline to this topic. In other words, features in very short texts are quite consistent. It is expected therefore that this kind of characteristic can be used properly, and we can improve the short text classification performance. CRFs can be used to add constraints among features.

CRFs are undirected graphical models trained to maximize a conditional probability first introduced by Lafferty, such as natural language processing to the indexing of the string to learning tasks.

However, CRFs cannot be used in short text classification directly, because it is a sequence labelling algorithm. Therefore, we need to convert short text classification problem to sequence labelling problem. In this section, we will propose our CRFs-based classifier. We borrow the character tagging approach, which is widely used in Chinese word segmentation task [18][19], to covert our classification problem to a sequence problem, so that CRFs can be used in our approach. Finally, we propose our novel short text classification method based on CRFs.

3.1 Proposed algorithm

In this subsection, we borrow the tagging approach to reformulate short text classification task as a sequence labelling problem.

This approach is used in both train and test step. Nevertheless, our algorithm tags every character with the category of the short text. We will use an example to illustrate our proposed algorithm.

Assume we have such a short text in train set,

Uyghur characters: ئۈچتە ياخشى ئوقۇغۇچىلار تەقدىرلەندى

Uyghur Latin characters: Üchte Yaxshi Oqughuchilar Teqdirlendi

("Three good" student was praised).

This text belongs to Education class.

Uyghur characters: بۇ يىل ئۆي باھاسى ئىككى پىرسەنت ئۆستى

Uyghur Latin characters: Buyil Öy Baxasi Ikki Pirsent Östi

(House prices rose two percent in this year).

This text belongs to Business class.

Uyghur characters: خەلقئارا سەھىيە تەشكىلاتى زۇكامغا قارشى
ۋاكسىنىنى نامرات دۆلەتلەرگە ئەۋەتتى

Uyghur Latin characters: Xeliqara Sehiye Teshkilati Zukamgha Qarshi Waksinini Namrat Döletlerge Ewetti

(WHO send swine flu vaccine to poor countries).

This text belongs to Health class.

Uyghur characters:
چېگرادىن چىقىرىلغان ئۆزبەكلەر قىرغىزتانغا قايتۇرلىشتىن ئەنسىرەيدىغانلىقىنى
بىرلەشكەن دۆلەتلەرتەشكىلاتىغا خەۋەر قىلدى

Uyghur Latin characters: CHëgradin CHiqirilghan Özbekler Qirghiztangha Qayturlishtin Ensireydhghanlighini Birleshken Döletler Teshkilatigha Xewer Qildi

(The deported Uzbek people told the UN fears returned to Kyrgyzstan).

This text belongs to Politics class.

Uyghur characters: بارسېلۇنالىق مېسسىي بۇ يىل دۇنيا توپ چولپىنى دېگەن نامغا ئېرىشتى

Uyghur Latin characters: Barsëlunaliq Mëssiy Bu Yil Dunya Top CHolpini Dëgen Namgha Ërishti

(Barcelona's Messy named World Player of the Year).

This text belongs to Sports class.

Uyghur characters: جېكسوننىڭ ئۆلىمى دۇنيانى تەۋرەتتى

Uyghur Latin characters: Jëksonning Ölimi Dunyani Tewretti

(Jackson's death shocked all over the world).

This text belongs to Play class.

If we turn the short texts with category to labeled sequence, CRFs algorithm can be used to train models, which can be used in test step.

Assuming we have such a short text in test set,

Uyghur characters: ئىككى تالىبان ئافغانىستاندا ئۆلدى

Uyghur Latin characters: Ikki Taliban Afghanistanda Öldi

(Two Taliban dead in Afghan city gun battle).

This text belongs to Military class. This short text is turned into a labeled sequence as shown in Fig. 1.

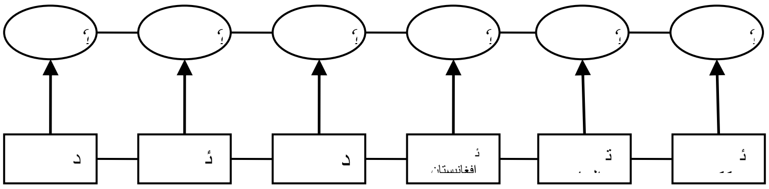


Fig. 1. Graphical structures of linear chain

In Fig. 1, square nodes indicate characters and circle nodes indicate tags according to the characters. We will use CRFs algorithm to infer its category.

We use the CRFs model obtained in the train step to infer the category of short text in test set.

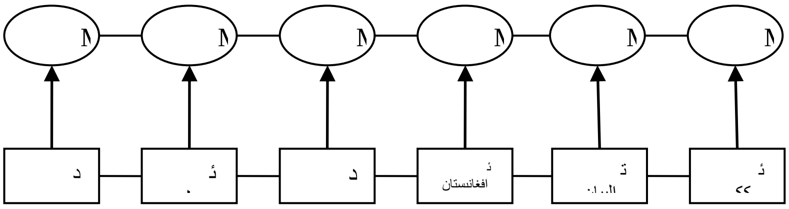


Fig. 2. Graphical structures of linear chain

In Figure 2, we get the category of the short text. This short text belongs to “Military” class.

Our analysis shows that tokenization changes under different lengths of texts, has a slight negative correlation with the length.

The problem of short text classification can be formally stated as follows: Given a sequence of token $w_i \ldots w_n$, we want to find the corresponding sequence of classification tags $t_1 \ldots t_n$, drawn from a set of tags T, which satisfies:

$$P(t_1 \ldots t_n | w_1 \ldots w_n) = \prod_{i=1}^{n} P(ti \mid Ci) \tag{1}$$

where Ci is the context for token w_i.
The best feature templates used in our short text classification experiments:
Unigram features:
U_{i-1}: previous word in sentence
U_i: current word in sentence
U_{i+1}: next word in sentence
Bigram features:
$U_{i-1}U_i$, U_iU_{i+1}

We incorporate morphological information in Uyghur by using a morpheme analyzer [20].The features of very short texts tend to be consistent. From the graphical structure of linear chain CRFs, we can find constraints among the hidden variables. In other words, there are constraints among the topics. This method calculates similarities between two strings (e.g. texts or sequences) by matching the common substring in the strings. Therefore, our CRFs-based approach is expected to yield promising performance.

The main idea of the algorithm is to select feature words from each document; those words cover all the ideas in the document. The results of this algorithm are a list of the main subjects founded in the document. Also, in this paper the effects of the Uyghur text classification on Information Retrieval have been investigated. The goal was to improve the convenience and effectiveness of information access. We will quantitatively evaluate our method in the next section.

4 Experiments

In the data preparation step, only long titles are eliminated, noisy text such as stop words are still in our dataset. Therefore, the accuracies of both classifiers are relatively low. Nevertheless, our CRFs-based approach still outperforms SVM classifier. Three different datasets are used to evaluate our CRFs based approach.

Uyghur datasets compose titles of web documents. Sogou[2] short text corpora are Chinese datasets, which compose of titles of web documents. Ohsumed-all dataset is an English corpus, which consists of medical abstracts.

In our experiments, unigram and bigram features are used. On the other hand, we focus on very short texts. Therefore, the lengths of all the texts we used are less than twelve. It is just an empirical value, because there is no clear definition of “very short text”. In Uyghur texts, the limit is twelve token and In Chinese texts, the limit is twenty characters, while in English texts, and the limit is ten words.

In the next subsections, we compare our CRFs-based approach with SVM classifier in the evaluation forms. Libsvm[3] is used as our basic classifier as it has been proved to be effective on many machine learning tasks especially text classification. We use probability SVM models in this paper.

4.1 Uyghur Short Text Classification

Uyghur Energy（UE）corpus is a relatively small dataset. Therefore, we conduct experiments on it to verify our results. UE short text corpora are used in this subsection.

Uyghur titles of web documents corpus has seven subject categories and 59,992 single-labeled documents. We use the titles of web documents as a short text corpus.
All the short texts are used in seven class classification experiments. "sports", "military", "play", "health", "politics", "education", "business" classes are used in our experiments. The whole corpus is randomly divided into train set and test set with different proportion.

The classifying accuracy and train time (second) comparisons according to different proportions of train set and test set (7.5:2.5, 5:5, 3:7) are shown Figure3-5 (Table 1-3) and Table 4-5 as follows:

[2] http://www.sogou.com/labs/resources.html

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

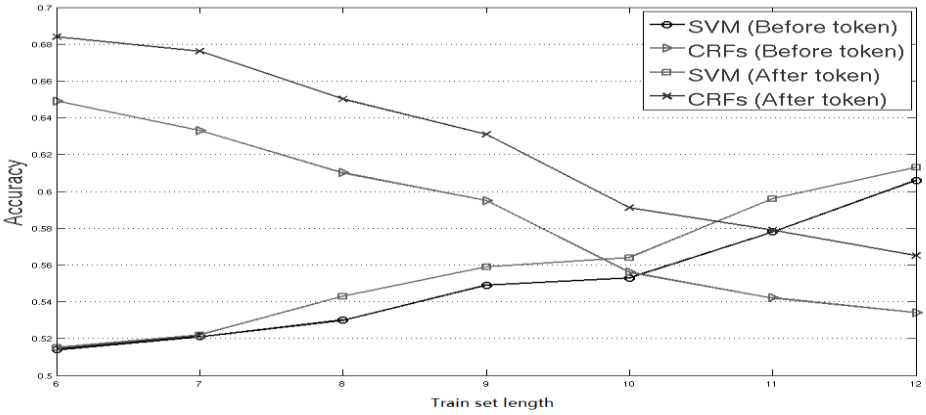


Fig. 3. The classifying accuracy comparisons on UE corpus (train: test=7.5:2.5)

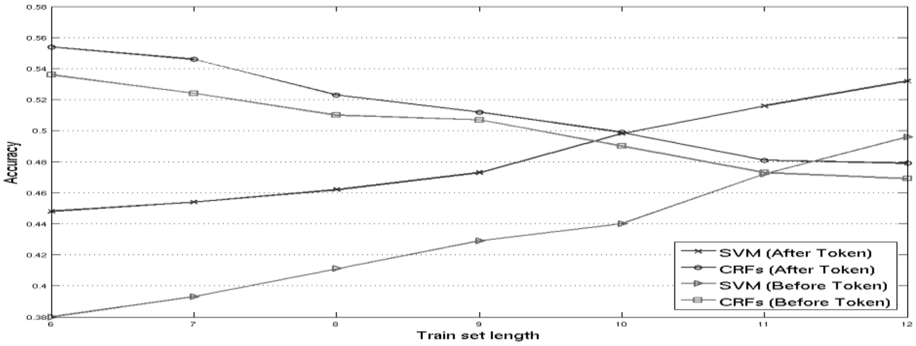


Fig. 4. The classifying accuracy comparisons on UE corpus (train: test=5:5)

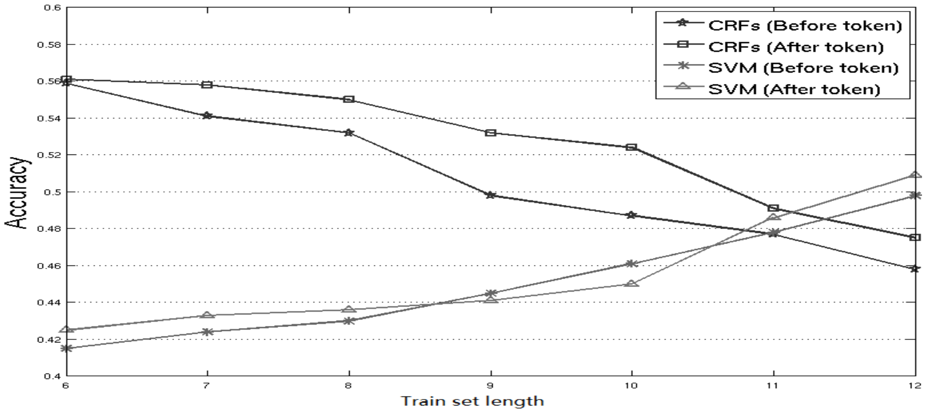


Fig. 5. The classifying accuracy comparisons on UE corpus (train: test=3:7)

Table 1: The classifying accuracy on UE corpus (train: test=7.5:2.5)

Train set length	MEM (Before token)	MEM (After token)
6	51.22	54.84
12	66.83	67.79

Table 2: The classifying accuracy on UE corpus (train: test=5:5)

Train set length	MEM (Before token)	MEM (After token)
6	42.11	41.27
12	54.78	53.96

Table 3: The classifying accuracy on UE corpus (train: test=3:7)

Train set length	MEM (Before token)	MEM (After token)
6	40.58	43.55
12	54.09	56.10

Table 4: The train time comparisons on UE corpus (train: test=7.5:2.5)

	Before Token	
Train set length	SVM	CRFs
6	14100	2400
12	16140	4260
	After Token	
Train set length	SVM	CRFs
6	14820	8760
12	16440	9900

Table 5: The train time comparisons on UE corpus (train: test=5:5)

	Before Token	
Train set length	SVM	CRFs
6	5400	1740
12	5880	3000
	After Token	
Train set length	SVM	CRFs
6	5580	4080
12	6240	5880

Experimental results in the tables above demonstrate our CRFs-based approach can yield promising performances. Two different datasets are used to evaluate our CRFs based approach.

UE and UE-tokenized short text corpora are Uyghur datasets, which are compose of titles of web documents.

On the experiments on UE corpus, our proposed CRFs-based approach outperforms SVM classifier and MEM classifier.

4.2 Chinese short text classification

Sogou web documents corpus is a public dataset. We use the titles of web document as a short text corpus. After eliminating long titles, we finally obtain 14 subject categories (sports, military, play, health, politics, education, business, women, culture, house, news, information, education, travel, and auto) and 40894 single-labeled documents.

Previous work [21] shows that Chinese character bigram has better performance than Chinese word unit. Besides, we don't need to take Chinese word segmentation into consideration. The classifying accuracy comparisons according to different proportions of train set and test set (7:3) are shown Table 6 as follows:

Table 6: The classifying accuracy comparisons on Sogou corpus

Train set length (characters)	SVM	CRFs
10	0.693	0.745
20	0.791	0.801

Experimental results in the tables above demonstrate our CRFs-based approach still outperforms SVM classifier. Our method is also valid on Chinese datasets.

4.3 English short text classification

We also conduct experiments on English corpora. Ohsumed (MEDLINE) is used in this subsection. Ohsumed-all dataset is composed of 50216 medical abstracts classified into 23 categories (C1,C2…C23). After eliminating long titles (longer than ten words), we finally obtain 28399 very short texts in 23 different classes. We conduct experiments on 4 class classification and 5 class classification tasks.

Table 7: Performance comparisons on Ohsumed corpus (four class classification)

	SVM	CRFs
C1~C4	0.560	0.627
C5~C8	0.457	0.494
C9~C12	0.501	0.564
C13~C16	0.590	0.619
C17~C20	0.432	0.475

Table 8: Performance comparisons on Ohsumed-all corpus (five class classification)

	SVM	CRFs
C1~C5	0.501	0.558
C6~C10	0.431	0.479
C11~C15	0.501	0.531
C16~C20	0.382	0.421

In the data preparation step, only long titles are eliminated, noisy text such as stop words are still in our dataset.

Therefore, the accuracies of both classifiers are relatively low. Nevertheless, our CRFs-based approach still outperforms SVM classifier. Our method is also valid on English datasets.

Experiment results demonstrate that our CRFs-based approach can yield promising performances. CRFs-based short text classification approach outperforms SVM classifier on all datasets.

5 Conclusion

In this paper, we presented a brief overview of the text classification task. We applied two supervised learning algorithms, SVM and CRFs, for Uyghur, Chinese and English short text classification. We also compared their performance with different proportion of train set and test set.

This paper presents enhanced, effective and simple approach to short text classification. The approach uses an algorithm to automatically classify documents.

The main idea of our work is that features have strong relationship in very short texts. The features of very short texts tend to be consistent. On the other hand, from the graphical structure of linear chain CRFs, we can find constraints among the hidden variables. In other words, there are constraints among the topics. Therefore, our CRFs-based approach is expected to yield promising performance.

Three different datasets are used to evaluate our CRFs based approach and that both datasets are used in our evaluation section.

Our method can greatly decrease error rates, particularly if the number of examples or the size of the strings in the training set is small. All experiment results demonstrate that our proposed algorithm can yield stable and significant improvements on different kinds of classification tasks. Therefore, our approach can also be used in regular text classification task.

In comparison with large train languages, such as Chinese and English, our system has less performance. This could be caused by the limited number of train words which are not enough to cover the UE. But our research will be very useful in the development of Turkic language families (include Turkish, Azeri, Uzbek, Kazakh, Turkmen, Tatar, Kyrgyz and others) text classification and other related research. How to use it properly will be our future work.

In future, we plan to conduct more experiments on SVM and CRFs with multi-class documents, which is a large number of single class documents (include micro blogs) and multi label documents.

Acknowledgements

We would like to thank anonymous reviewers for their useful comments and suggestions. This work is supported by Xinjiang University doctoral project.

References

1. Batuer Aisha , Maosong Sun.: A Statistical Method for Uyghur Tokenization, In Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2009,pp.383-387.
2. Eibe Frank and Remco R. Bouckaert.: Naïve Bayes for text classification with unbalanced classes. In Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer, Berlin, Germany, 2006, pp.503-510.
3. John Lafferty, Andrew McCallum, and Fernando Pereira.: Conditional random fields: probabilistic models for segmenting and labeling sequence data, In Proceedings of ICML, 2001, pp. 591-598.
4. T. Joachims.: Text categorization with Support Vector Machines: Learning with many relevant features, In Proceedings of the 10th European Conference on Machine Learning, 1998, pp. 137-142.
5. A. Berger, S. Della Pietra, and V. Della Pietra.: A maximum entropy approach to natural language processing. Computational Linguistics, 1996, 22(1):39-71.
6. F.Sebastiani.: Machine learning in automated text categorization, ACM Computing Surveys, 2002, 34(1):1- 47 .
7. Yiming Yang, Xin Liu.: A re-examination of text categorization methods, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, California, United States, August 15-19, 1999, pp.42-49 .
8. Zelikovitz, S. and Hirsh, H.: Improving short-text classification using unlabeled background knowledge to assess document similarity, Proceedings of the Seventeenth International Conference on Machine Learning (2000) .
9. Shen, D., Pan, R., Sun, J.-T., Pan, J. J.;Wu, K., Yin, J., and Yang, Q.: Query enrichment for web-query classification, ACM Trans. Inf. Syst. 2006, 24(3):320-352.
10. M. Sahami and T. Heilman.: A Web based kernel function for measuring the similarity of short text snippets, In Proc. WWW (2006).
11. D. Bollegala, Y. Matsuo, and M. Ishizuka.: Measuring semantic similarity between words using Web search engines, Proc. WWW (2007).
12. W. Yih and C. Meek.: Improving similarity measures for short segments of text, In Proc. AAAI (2007).
13. Huanhuan Cao, Derek Hao Hu and Dou Shen et al.:Context-Aware Query Classification, Proc. ACM SIGIR (2009).
14. JS. Banerjee, K. Ramanathan, and A. Gupta.: Clustering short texts using Wikipedia, Proc. ACM SIGIR (2007).
15. X. Phan, L. Nguyen, and S. Horiguchi.: Learning to Classify Short and Sparse Text and Web with Hidden Topics from Large-scale Data Collections, In Proc. WWW (2008).
16. P. Schonhofen.: Identifying document topics using the Wikipedia category network, Proc.

the IEEE/WIC/ACM International Conference on Web Intelligence, (2006).

17. Jan Blaˇt´ak, Eva Mr´akov´a andLuboˇs Popel´ınsk´.: Fragments and Text Categorization, Proceedings of the ACL 2004.
18. Fuchun Peng, Fangfang Feng, and Andrew McCallum.: Chinese segmentation and new word detection using conditional random fields, In Proceedings of the 20th international conference on Computational Linguistics, 2004,pp. 562-568.
19. Xue. Nianwen: Chinese word segmentation as character tagging, In Journal of Computational Linguistics and Chinese Language Processing, 2003, 8(1): 29-48.
20. Batuer Aisha, Maosong Sun, A Uyghur Morpheme Analysis Method based on Conditional Random Fields, International Journal on Asian Language Processing, 19(2),2009,69-77.
21. J.Y. Li, Mao song Sun, Xian Zhang.: A Comparison and Semi-Quantitative Analysis of Words and Character-Bigrams as Features in Chinese Text Categorization, In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 2006, pp. 545-552.

Fast and Accurate Language Detection in Short Texts using Contextual Entropy

Edgar Chávez[1], Moisés García[2], Jesus Favela[1]

[1] CICESE, Ensenada, México {elchavez,favela}@cicese.mx
[2] Universidad Michoacana, México moises@fie.umich.mx

Abstract. In this work we address the problem of Language identification (LI) on short segments of text. The central idea is to compute the entropy of a document in different contexts and assign it to the category where the entropy is maximal. Only word distributions are needed for the task, no other training is done. For LI the contexts are the languages, and classification is done by just evaluating the high order entropy of the text. Our results show that the language of the text, in the challenging case of short texts, can be accurately identified, matching state of the art approaches reported in the literature. Our method is also fast, given its simplicity, it is easy to code and needs no training, aside from the estimation of words distributions for each language, if not already available.

1 Introduction

The information in social media is growing exponentially. An important part of this corpus is made of short texts, these messages tend to be written in an informal tone, using different languages and are often not grammatically correct.

The problem of language identification (LI) is interesting on its own. However, in many practical applications LI can be seen as step of a more complex text processing task. Accurate LI can facilitate the use of background information about the language and the use of more specialized natural language processing approaches dealing with a collection or a stream of texts [10].

There is widespread misconception of LI being a "solved task", generally as a result of isolated experiments over homogeneous datasets[5, 12]. Part of the motivation for this paper is to draw attention to the fact that, as a field, we are still a long way off from perfect LI of web documents, mainly due to the small size of the texts and the number of languages available on the Web.

The popularity of social media, including Twitter and social networking sites, has generated research on social media data analysis, such as opinion mining. Accurate LI on short and grammatically-ill text messages is thus, required to support advances in these areas.

We consider LI as a text classification problem –the assignment of natural language texts to one or more predefined categories (languages in this case) – Text classification poses many challenges for inductive learning methods since there can be millions of word features. The resulting classifiers, however, have

pp. 351–358; rec. 2015-01-31; acc. 2015-02-27

many advantages: they are easy to build and update, they depend only on information that can be easily obtained (i.e., samples of items that are within the same categories), they can be customized to specific categories of interest to individuals, and they allow users to smoothly tradeoff precision and recall depending on their task.

In this work, we propose the use of an approach for text classification based on the Contextual Entropy (CE) [4], with a modification that considers that all words in each category (language) have equal probability. Thus, we compute the contextual entropy value using different word distributions, one for each document class, then classify the document as belonging to the class which maximizes the entropy (or a higher moment of it).

The rest of the paper is organized as follows. In section 2 we define the problem of language identification. In section 3, we define and explain contextual entropy. Section 4, presents experimental results comparing the performance of our approach to LI with previous results. Finally, section 5 contains our conclusions and suggestions for future work.

2 Language Identification

LI is a supervised learning task, particularly a plain single label multi-class classification [10]. Given some historical or training data in which for each text t there exists a label l, the language in which this text is written, the goal is to learn a model such that given some previously unseen text, it can identify, as accurately as possible, in which language this text is written. Some cases to classify text in LI are: a) When a text written partly in one language and partly in some other language and someone would like to get both labels as an output. b) Language groups or any other dependencies between the language labels. Case (a) can be solved by chopping the text in small portions (for example by segmenting the text in phrases and classifying each phrase independently). Case (b) refers to languages that have many common words, like italian, spanish, portugues, etc. Both problems can be solved by aiming at short text analysis, as we do in this paper.

A recent study [11] established that the best known technique to address this problem is to classify documents according to rank order statistics over character n-gram sequences between a document and a global language profile [1]. Other statistical approaches applied to LI include Markov models over n-gram frequency profiles [3], dot products of word frequency vectors [2], and string kernels in support vector machines [7].

In contrast to purely statistical methods, linguistically-motivated models for LI have also been proposed, such as the use of stop word lists [6], where a document is classified according to its degree of overlap with lists for different languages. This latter idea is taken into account in the development of our approach.

3 Contextual Entropy

In previous work [4] we have shown that entropy is effective in text categorization for formal as well as for intuitive reasons. The entropy is a statistic that depends both on the object itself (the text) and the context (the vocabulary distribution). This fact is strong, specially because it accounts for the property we are seeking: Putting the text in the right context. Formally measuring the amount of information in the process of communication was introduced by Shannon's work [9], the amount of information in the text can be determined by the value of entropy, according to equation 1.

$$H(X) = -\sum_i p_i log(p_i) \tag{1}$$

Where X is a discrete variable, and $\{p_i\}$ are the probability of taking the value i. The above equation 1 plays a central role in information theory as a measure of information, choice and uncertainty [8]. Shannon suddenly jumps from the description of information production to information itself, choice, and uncertainty. H measures the amount of information transmitted in the communication process as a product of the selection of one out of several possible messages, reflecting the occurrence of one event out of several events in the information source.

Higher order moments of the entropy has been shown to be more accurate for the task, and Contextual Entropy, defined according to equation 2 [4] optimizes for a certain value of n. This optimal classification value need to be estimated empirically.

$$H(c)(X) = -\sum_i p(c)_i \log^n(p(c)_i) \tag{2}$$

In equation 2, $H(c)$ is the entropy computed in context c, and X is a a document, $p(c)_i$ is the probability of the word i in context c. The value n is empirically obtained. The document is assigned to the language maximizing the higher order entropy. In figure 1 the entropy with highest value computed correspond to the distribution that belongs the test document.

To show how the value n affects the computation of equation 2, consider that each element in the $\mathbf{W}$ distribution is equally probable, i.e. have the same value given by $p = \frac{1}{|W|}$. If we have multiple distributions ($\mathbf{W}_m$) and they give close entropy values, when we increase n and compute H the net effect is an increase in the gap values. Figure 2 shows the effect of the computed values for H for various values of n . The results better for some specific values of n.

3.1 Classification

The classification can be speed up with a rather simple procedure. We only need to verify common/unique words in each one of the distributions and compute the cumulative contextual entropy accordingly. Lets see the details. Let W_A be

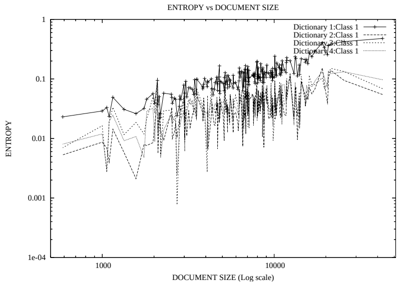


Fig. 1. Entropy computed using training documents of the class k and different distributions $\mathbf{W}_m$. The distributions were obtained from m classes (1 to 4) of the training data as example.

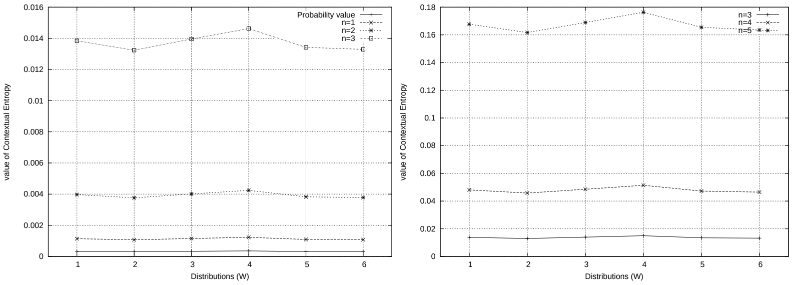


Fig. 2. Increasing n from equation 2.

the distribution of class A, and let W_B be the distribution of class B, such that $P_i = \{p_1, p_2, \ldots, p_d\}$ represent the probability distribution of the words w_d obtained from class i, $i = \{A, B\}$, in our approach all p_d are equal, for $0 \leq p_d \leq 1$, d is the number of words in the dictionary W, and

$$\sum_d p_d = 1$$

Let a new document $\mathbf{x} = \{x_1, x_2, \ldots, x_l\}$ where x_l is the word w_d in the document $\mathbf{x}$. First we compute the entropy using P_A and P_B.

$$H_A(\mathbf{x}) = \sum_l p_l log^n(p_l)$$

$$H_B(\mathbf{x}) = \sum_l p_l log^n(p_l)$$

Then we obtain the $max(H_i)$ and decide the language to which the new document belongs. Figure 3 shows a flow diagram that summarizes the process for LI.

The computational complexity for training the model in our approach is given by $O(c|D|)$, where c is the number of clases, and $D = \{D_1, D_2, \ldots, D_k\}$ denote

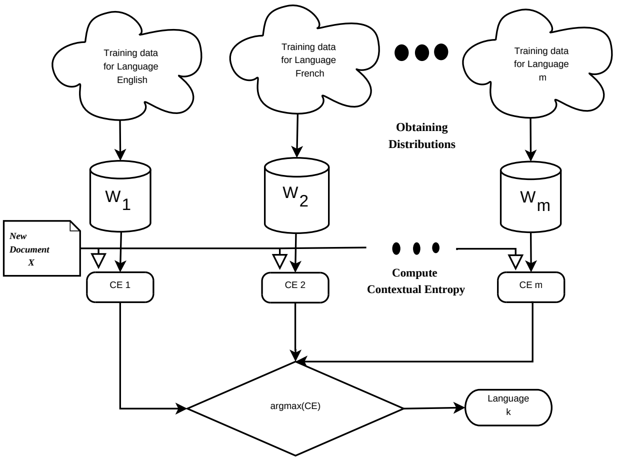

Training data for Language English
Training data for Language French
Training data for Language m
Obtaining Distributions
W1
W2
Wm
New Document X
CE 1
CE 2
CE m
Compute Contextual Entropy
argmax(CE)
Language k

Fig. 3. The CE of document **x** is calculated for each distribution W_i or information sources in the system. The class k with highest entropy coincides with to the document **x** class.

a training set of k documents where each document has been assigned a label from the classes label set. To classify a new document **x**, we use only the words of the document **x** to compute the entropy in each class c, so the classifcation complexity is $O(cL(x))$, with $L(x)$ the length of the document to be classified.

4 Experimental Results

4.1 Data Sets

The dataset provided by [10], consist of 9066 tweets for 6 languages of at most 140 bytes. The six languages are English, French, German, Spanish, Dutch and Italian. These are languages we have sufficient knowledge of to identify. Note that Spanish, French and Italian presents a challenge as these languages contain a lot of similar word extensions.

4.2 Experiments

In order to compare the accuracy of the methods reported in [10], we train a model with one part of the data called train. We vary the size of the train data to establish the influence of the corpus' size. We use 5%, 10%, 25% and 50% of the entire dataset stratified per language and sampled uniformly.

In all cases we compare the mean accuracy of the CE for $n = 2$ and $n = 5$. The mean accuracies are obtained by taking the mean of 50 different repeated random sub-sampling validation experiment runs.

The main results are presented in table 1 which shows, averaged over 50 experiments runs, the accuracies and standard deviations for CE using $n = 2$ and $n = 5$, LIGA and N-gram reported by [10].

Table 1. Accuracy averaged over 50 runs.

Train Docs.	LIGA	N-gram	CE n=2	CE n=5
5%	94.9 ± 0.8	87.5 ± 1.5	97.1 ± 2.5	97.3 ±2.0
10%	96.4 ± 0.5	90.6 ± 1.0	97.9 ± 1.3	98.2 ±0.5
25%	97.3 ± 0.5	92.5 ± 0.9	98.6 ± 0.9	99.0 ±0.4
50%	97.5 ± 0.5	93.1 ± 0.8	98.9 ± 0.5	99.2 ±0.1

To show the performance of the classifier respect to n, another experiment was carried out using 50% of the documents for training and the rest to test the classifier at different values of n in the equation for entropy contextual. Table 2 and figure 4 present the averaged accurate of this experiment.

Table 2. Accuracy averaged for CE over 20 runs for 50% of documents for trainig at diferent values of n.

n value	Accuracy	Variance
2	98.92436	0.53479
3	99.09331	0.26163
4	99.11911	0.19739
5	99.22973	0.14092
6	99.24737	0.15320
7	99.29478	0.17751
8	99.25101	0.18417
9	99.09037	0.41025
10	99.11566	0.45383
11	99.15769	0.34583
12	99.11285	0.43935
13	99.02434	0.57851
14	98.75165	0.96047
15	98.44259	1.65902

In figure 4 we can see the behavior of CE classifier to different values of n, that as in [4] the best results are obtaining on the first values of n.

5 Conclusions and future work

We presented a new approach for language identification using contextual entropy. Our empirical analysis shows that CE is better than state of the art tech-

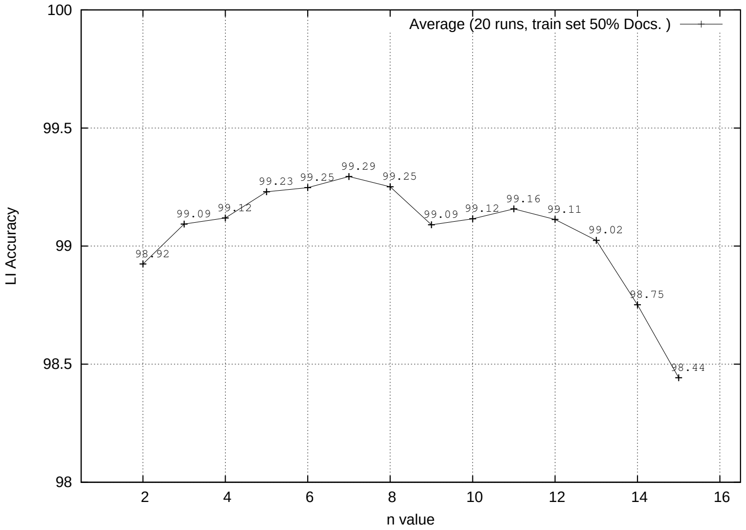


Fig. 4. Average accurate at some values of n to the CE classifier.

niques used in LI. Even more interesting is the change of perspective, seeking the overlap in the distribution that considers each word equally probable.

The computational complexity for LI of a new document using CE is $O(cL(x))$, where c is the number of classes, and $L(x)$ is the length of the document to be classified.

Our preprocessing step consists only in obtaining the different words in the training set to generate the distribution of each language and count the words, we don't need another text operation.

We are currently working in filtering the vocabulary to increase the intra-class entropy as well as decreasing the inter-class entropy. We are also working on other applications of text categorization that may benefit from our method, such as sentiment analysis, mood detection and language identification in multilingual documents.

References

1. William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.
2. Marc Damashek et al. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848, 1995.
3. Ted Dunning. *Statistical identification of language*. Computing Research Laboratory, New Mexico State University, 1994.
4. Moises Garcia, Hugo Hidalgo, and Edgar Chavez. Contextual entropy and text categorization. In *Web Congress, 2006. LA-Web'06. Fourth Latin American*, pages 147–153. IEEE, 2006.

5. Baden Hughes, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew Mackinlay. Reconsidering language identification for written language resources. In *Proceedings of LREC2006*, pages 485–488, 2006.
6. Stephen Johnson. Solving the problem of language recognition. Technical report, University of Leeds, 1993.
7. Canasai Kruengkrai, Prapass Srichaivattana, Virach Sornlertlamvanich, and Hitoshi Isahara. Language identification based on string kernels. In *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, volume 2, pages 926–929. IEEE, 2005.
8. Marcin J. Schroeder. An alternative to entropy in the measurement of information. *Entropy*, 6:388–412, 2004.
9. Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, 1948.
10. Erik Tromp and Mykola Pechenizkiy. Graph-based n-gram language identification on short texts. In *Proceedings of the Twentieth Belgian Dutch Conference on Machine Learning (Benelearn 2011)*, pages 27–34, 2011.
11. NICTA VRL. Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 17–25, 2014.
12. Fei Xia, William D. Lewis, and Hoifung Poon. Language id in the context of harvesting language data off the web. In *Proceedings of the 12th Conference of the EACL (EACL 2009)*, pages 870–878, 2009.

Staged Approach for Grammatical Gender Identification of Nouns using Association Rule Mining and Classification

Shilpa Desai[1], Jyoti Pawar[1], and Pushpak Bhattacharyya[2]

[1] Department of Computer Science and Technology
Goa University, Goa - India
sndesai@gmail.com, jyotidpawar@gmail.com
[2] Department of Computer Science and Engineering
IIT-Bombay, Mumbai - India
pb@iitb.ac.in

Abstract. In some languages, gender is a grammatical property of the noun. Grammatical gender identification enhances machine translation of such languages. This paper reports a three staged approach for grammatical gender identification that makes use of word and morphological features only. A Morphological Analyzer is used to extract the morphological features. In stage one, association rule mining is used to obtain grammatical gender identification rules. Classification is used at the second stage to identify grammatical gender for nouns that are not covered by grammatical gender identification rules obtained in stage one. The third stage combines the results of the two stages to identify the gender. The staged approach has a better precision, recall and F-score compared to machine learning classifiers used on complete data set. The approach was tested on Konkani nouns extracted from the Konkani WordNet and an F-Score 0.84 was obtained.

1 Introduction

Gender is a grammatical property of nouns in many languages[3] including Indian language such as Sanskrit, Hindi, Gujarati, Marathi and Konkani. In such languages adjectives and verbs in a sentence agree with the gender of the noun. For example translation of "*He is a good boy*" and "*She is a good girl*" in Hindi is "*vaha eka achchhaa laDakaa haai*[4]" and "*vaha eka achchhii laDakii haai*", respectively. We see that adjective good is translated into "*achchhaa*" or "*achchhii*" based on the gender of the noun. Using a freely available and highly used Google Translate machine translation system, translation of "*It is a nice river*" to Hindi is "*yaha eka achchhaa nadai haai*". The grammatical gender

[3] List of languages with type of grammatical gender followed is listed at (http://en.wikipedia.org/wiki/List_of_languages_by_type_of_grammatical_genders)

[4] In the paper we have used transliteration for Hindi and Konkani examples that resembles ITRANS (http://www.aczoom.com/itrans/html/tblall/tblall.html).

pp. 359–371; rec. 2015-01-25; acc. 2015-02-27

of river i.e. "*nadai*" in Hindi is feminine hence "*achchhii*" should be used in place of "*achchhaa*" for the translation to be more appropriate. Thus, machine translation into any language with a grammatical gender will need to predict the grammatical gender of the noun to ensure agreement. Hence, determining grammatical gender of nouns in these languages is essential.

Features such as grammatical gender of nouns required for machine translation can be obtained from a lexical resource like WordNet. However all WordNets do not maintain gender of nouns. Many WordNets in Indian Languages namely Marathi, Konkani, Gujarati, Bengali, Sanskrit etc. have been created using the Expansion Approach[1] with Hindi WordNet as source. Hindi nouns have either *masculine* or *feminine* grammatical gender whereas Bengali nouns do not possess grammatical gender. Nouns in Marathi, Konkani, Gujarati and Sanskrit can have either *masculine* or *feminine* or *neuter* grammatical gender. Grammatical gender of nouns was not directly borrowed from the source WordNet as all language do not share the same grammatical gender. There is a need to augment the WordNet nouns with grammatical gender so that WordNet can be a valuable resource for machine translation, specially for languages which have an agreement system with another aspect of the language, such as adjectives, articles or verbs.

In this paper we present an automatic three stage gender identification approach using word and morphological features, which can be used to assign grammatical gender to nouns in a WordNet. Our method was tested for nouns in Konkani WordNet[5] . The main contributions of this work can be summarized as follows:

- Grammatical Gender Identification rules obtained using association rule mining.
- Classification model for grammatical gender identification where association rule found insufficient.
- New grammatical gender category called non-neuter[6]

The rest of the paper is organized as follows: Section 2 is devoted to related work. A description of grammatical gender identification is discussed in section 3. The staged approach used for grammatical gender identification is described in section 4. In Section 5, experimental results and evaluation are discussed. Section 6 concludes the paper.

2 Related Work

Related work for lexicon extension include semi-supervised approach used to predict lexical properties like grammatical gender for Wambaya[2], a low density language. Here co-occurrence frequencies with demonstratives and machine

[5] The Konkani WordNet can be accessed at http://konkaniwordnet.unigoa.ac.in/public/wordnet/wordnet.php?langid=19&id=2

[6] Discussed in section 3

learning using context windows are used to predict the lexical properties. Gender classification, for Dutch nouns, has been attempted for rule-based MT generation tasks[3]. In which information derived from large annotated monolingual corpora, a set of context-checking rules based on co-occurrence of nouns and determiners are used. An SVM based tool has been developed for Romanian[4] to distinguish neuter gender which uses n-gram features.

General sound of the language has been exploited to determine the grammatical gender for German and Romanian[5]. Noun's ending have been used as a strong clue for gender in French[6]. Cognitive aspects of grammatical gender for Italian nouns referring to animals have been tested[7]. Bootstrapping process which make use of word forms, determiners, quantifiers and adjectives to predict gender for nouns in context[8] have been carried out.

Other reported work on gender identification has been for anaphora resolution wherein gender information has been extracted from enhanced part-of-speech tagger[9]. Methods to classify all occurrences of nouns in a document using a wide variety of contextual, morphological and categorical gender features have been done[10]. These works are targeted towards biological gender identification and not grammatical gender.

Most of the work reported is for non-Indian languages and make use of determiners or gender-marked pronouns like his, her etc. Some of these methods are used for gender identification within a context. Other methods require manual preparation of annotated data like morphological suffix list with corresponding gender which is time consuming.

3 Grammatical Gender Identification

Noun-class[11] in linguistics refers to a particular category of nouns. Some natural languages categorize nouns into noun-classes based on either semantic criterion, morphology or some arbitrary convention. Noun-classes form a system of grammatical agreement. Some examples of common criteria that define noun classes include:

- animate vs. inanimate
- rational vs. non-rational
- human vs. non-human
- masculine vs. feminine
- masculine vs. feminine vs. neuter
- common vs. neuter

Grammatical gender[12] is a specific form of noun-class system. In languages with grammatical gender, every noun inherently carries one value (*masculine* or *feminine* or *neuter*). The values present in a given language are called the genders of that language[7]

[7] The grammatical gender for a noun mostly coincides with natural or biological gender but not always. For example "*chalii* (girl) in Konkani, "*Madchen* (girl) in German have grammatical gender as *neuter* and natural or biological gender as *feminine*

In Indian languages such as Sanskrit, Marathi, Konkani and Gujarati, the grammatical gender property of nouns is one of the gender values namely *Masculine*, *Feminine* or *Neuter*. Earlier profession names (job titles) like doctor, lawyer, professor, “*aadaogada* (advocate) etc were assigned *Masculine* gender because such professions were only practiced by men. However with changing times we see many women taking up such profession. Profession names now be used in either feminine or masculine sense and gender value can only be determined in the context of usage. For example “*Professor baraii shaikayata*” can be translated as ” *The professor teaches well*” or ” *She is a good professor*”. Here the noun “*Professor*” refers to a female professor which we can induce from the adjective agreement “*baraii*” in the sentence. However, the example “*Professor barao shaikayata*”, translated as ” *The professor teaches well*” or ” *He is a good professor*” where the noun “*Professor*” refers to a male professor which we can induce from “*barao*”. As a result such profession names cannot be assigned *Masculine* gender in languages like Konkani. To reflect this changing behavior in language a new gender value namely *Non-Neuter* was proposed. WordNet for languages like Konkani do not maintain two different entries for male Professor and female Professor, in such cases it is more relevant to assign *Non-Neuter* as grammatical gender value.

Definition *Non-Neuter* : *Non-Neuter* is a grammatical gender value assigned to a noun when the noun gender value can be either *masculine* or *feminine* based on the context and cannot be determined out of context.

3.1 Problem Statement

Given a set of nouns $N = \{n_1, n_2, ... , n_k\text{:}\ \ n_i$ is Konkani noun$\}$ and a set $W = \{(w_1,f_1), (w_2,f_2), ... , (w_m,f_m)$: $w_i \in$ Corpus C, f_i is the frequency count of w_i in $C\}$ the objective is to automatically identify appropriate grammatical gender for nouns in N and create set $N_G=\{(n_1,g_1), (n_2,g_2), ... , (n_k,g_k)$: $n_i \in N$, $g_i \in G\}$ where set $G = \{\ Masculine, Feminine, Neuter, Non\_Neuter\}$

```
Input   : Nouns
Output  : Gender label (Masculine, Feminine, Neuter or
          Non_Neuter).
Resources/tools used: Raw corpus, Morphological Analyzer and
                      WordNet.
Hypothesis: Morphological features along with word features can be
            used to identify grammatical gender of nouns.
```

To identify grammatical gender, we first identified all possible morphological and word features for nouns. We than prepared a data set based on these features. We describe the details of data set generation in the following sub-sections.

3.2 Data Set Generation

The features in the data set are grouped into two categories namely word features and morphological features. The morphological features are generated using the Konkani Morphological Analyzer[13]. The desired output feature, gender, is manually assigned for each tuple in the data set. The features used are listed in Table 1.

Let set α and set β hold all possible values for features BeginVowel and EndVowel respectively. To generate our data set we followed the following data preprocessing steps:-

Step 1: Extract nouns from the WordNet.

Step 2: Compute word features listed in Table 1 for each noun. Convert Unicode based string features to ASCII based string features using a map table.[8]

Step 3: Compute set α and set β

Step 4: For nouns extracted with each feature in set α and set β, randomly pick ten nouns of each word length wherever possible.

Step 5: Prune noun entries corresponding to morphological variants.[9]

Step 6: For each noun, compute values for morphological features in Table 1 using Morphological Analyzer tool for the language in this case Konkani.

Step 7: For each noun, manually assign gender feature.

We have a total of 1264 nouns in the data set used and the same can be found on http://www.cicling.org/2015/data/119/. The data set was used to identify a model for gender identification using supervised machine learning techniques. We ran various classification algorithms on our data sets[10] using 66% as training set and the rest as test set to determine the best training model. The performance of machine learning classifiers on our data set is presented in the next subsection.

3.3 Performance of Machine Learning Classifiers

The performance of various types of machine learning classifiers were tabulated based on word features only, morphology features only and both word and morphology features. The results[11] obtained for each feature type are tabulated in Table 2.

Analysis of Results: The performance of the various classifiers indicate that word + morphological features give better classification results compared to word features or morphological features used alone. The *ZeroR* Rule-Based classifier, which simply outputs the most common class irrespective of the attributes, is used as baseline or lower bound to evaluate classifier performance. *Bayesian*

[8] A map table assign a roman transliteration code to every Unicode characters. This mapping was done to execute machine learning classifiers which expect ASCII strings

[9] We found that some entries in WordNet were plural forms of the noun which were pruned

[10] Such data sets used to select a learning model are some times referred to as development (dev) set.

[11] Here Precision (P), Recall (R) and F-score (F) are the weighted average values generated

Table 1. Data Set Features.

Feature Name	**Feature Description**
Word Features	
Begin Vowel	The vowel attached to the letter with which the word begins. If the word does not have a vowel attached to first letter, then this feature is set to null.
End Vowel	The vowel if any with which the word ends. If the word does not end with a vowel, then this feature is set to null.
Length	The number of characters in the word.
Morphological Features	
PID	The matching paradigm identifier.
SOFS	The suffix attached to the noun base to get the singular oblique stem for the noun.
POFS	The suffix attached to the noun base to get the plural oblique stem for the noun.
PDCS	The suffix attached to the noun base to get the plural direct form of the noun.
FreqDSF	Number of times the direct singular form of the noun occurs in the corpus
FreqDPF	Number of times the direct plural form of the noun occurs in the corpus.
FreqSOF	Number of times the oblique singular form of the noun occurs in the corpus.
FreqPOF	Number of times the oblique plural form of the noun occurs in the corpus.
TotSOV	Total number of oblique singular forms of the noun occurring in the corpus.
TotalPOV	Total number of oblique plural forms of the noun occurring in the corpus.
SOVWin	Distinct number of types amongst oblique singular forms of the noun occurring in the corpus.
POVWin	Distinct number of types amongst oblique plural forms of the noun occurring in the corpus.
SR	Rule applied to obtain the noun stem.

Table 2. Performance of Machine Learning Classifiers.

	Features								
Algorithm		**Word**			**Morphological**			**Word + Morphological**	
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Bayesian									
Naive Bayes	0.676	0.679	0.658	0.770	0.781	0.766	0.775	0.784	0.769
Bayes Net	0.672	0.677	0.656	0.769	0.786	0.774	0.781	0.802	0.788
Function									
Logistic	0.696	0.712	0.689	0.788	0.809	0.792	0.786	0.805	0.794
MultilayerPerceptron	0.694	0.705	0.693	0.804	0.812	0.798	0.756	0.763	0.752
RBFNetwork	0.653	0.670	0.646	0.776	0.784	0.765	0.765	0.781	0.759
SimpleLogistic	0.709	0.691	0.661	0.774	0.805	0.786	0.792	0.821	0.803
SMO	0.721	0.681	0.644	0.775	0.807	0.787	0.783	0.816	0.798
Instance-Based									
IB1	0.713	0.714	0.697	0.757	0.767	0.759	0.774	0.781	0.766
KStar	0.728	0.714	0.697	0.747	0.753	0.744	0.761	0.763	0.754
Ensemble									
AdaBoost	0.416	0.563	0.437	0.605	0.576	0.467	0.496	0.626	0.542
Bagging	0.702	0.723	0.702	0.802	0.805	0.791	0.773	0.802	0.787
Random Sub Space	0.720	0.698	0.668	0.761	0.795	0.775	0.796	0.828	**0.811**
Decorate	0.688	0.714	0.695	0.801	0.814	**0.800**	0.802	0.819	0.804
Logit Boost	0.712	0.695	0.667	0.760	0.788	0.769	0.811	0.819	0.803
Rule-Based									
PART Decision List	0.701	0.723	0.700	0.770	0.786	0.776	0.790	0.809	0.798
Ridor	0.678	0.572	0.583	0.720	0.740	0.727	0.779	0.802	0.788
ZeroR	0.248	0.498	0.331	0.248	0.498	0.331	0.239	0.488	0.320
Decision Tree									
Random forest	0.726	0.726	**0.708**	0.777	0.791	0.779	0.784	0.809	0.794
Logistic Model Tree	0.709	0.691	0.661	0.774	0.865	0.786	0.799	0.826	**0.811**
REPTree	0.707	0.723	0.702	0.807	0.809	0.792	0.781	0.814	0.795

classifiers use a probabilistic framework, *Instance based classifier*s use nearest neighbor, distance based method for solving classification problem. These perform better than *ZeroR* the baseline used but don't provide the best learning model for the data. The Function based classifiers like *Logistic Regression, Multilayer Perceptron* fit a function on the attribute vectors in the training data to determine the class. They have been used as a good training model for many applications. However for grammatical gender identification, Logistic Model Tree classifier that combines logistic regression and decision tree learning performs better and is more appropriate learning model. Ensemble learning method namely Random Subspace which is a decision tree based classifier has a performance similar to Logistic Model Tree. These observations suggest that a combination of classifier provide a better model for grammatical gender identification. One amongst *Random subspace* and *Logistic Model Tree* can be used as a supervised machine learning model to identify grammatical gender with a F-Score of 0.811.

We were curious to know if we could extract some rules from our data which could be directly used to determine grammatical gender, and if so what is the precision or confidence level of such rules generated. Grammar books[14] and linguistic resources[15][16] available for the Konkani language do not explicitly specify any grammatical gender identification rules but suggest possibility of such rules. We wanted to check if the rules suggested and any new rule were generated by our data set and if so the precision or confidence level associated with such a rule. This motivated us to use a three stage grammatical gender identification approach described in the next section.

4 Staged Grammatical Gender Identification

4.1 Approach

A three stage approach was used wherein stage one used association rule mining to obtain a list of grammatical gender identification rules. Stage two used machine learning classifiers to identify grammatical gender of nouns that could not be identified in stage one using grammatical gender identification rules. In stage three the rules generated and best classification model obtained were combined to calculate the performance of the staged approach on a separate test data set with 200 entries. We describe the three stages used in the following sub-sections.

4.2 Stage One: Association Rule Mining

Predictive apriori association mining algorithm available in knowledge analysis tool WEKA 3.6.11 was used to generate gender identification rules. Three different experiments were conducted and rules were generated based on

1. Word features only
2. Morphological features only
3. Both word and morphological features.

For each type of feature multiple grammatical gender identification rules were obtained. Only those rules which had more than 0.95 confidence were accepted. Table 3 tabulates the number of rules extracted and some sample rules for each feature type. The rules obtained when both word and morphological features were used had some rules which were same as rules obtained using only word features and only morphological features. In addition some rules got combined to give more refined rules. Repetitive rules and refined rules were pruned and 16 rules were retained and implemented as Gender Identification Rule (GR).

Table 3. Grammatical Gender Identification Rules obtained.

Features Used	**# Grammatical Gender Rules**	**Sample Grammatical Gender Identification Rules Obtained**
Word	13	$EndVowel = o \rightarrow Gender = masculine$ acc:(0.99498) $EndVowel = ehm \rightarrow Gender = neuter$ acc:(0.99497) $BeginVowel = se\ EndVowel = a \rightarrow Gender = feminine$ acc:(0.99454) ......
Morphological	10	$PID = 20 \rightarrow Gender = masculine$ acc:(0.99496) $PID = 19 \rightarrow Gender = neuter$ acc:(0.99493) $PID = 13 \rightarrow Gender = feminine$ acc:(0.99476) ......
Word + Morphological	35	$PID = 13 \rightarrow\ EndVowel = be$ $Gender = feminine$ acc:(0.99469) $PID = 9 \rightarrow\ EndVowel = a$ $Gender = feminine$ acc:(0.99433) ......

The main contribution of using association rule mining is the 16 gender identification rules obtained with confidence above 0.95 for each rule. These rules alone can be used to determine gender of 49.76% of entries in the data set with a precision of 0.97. The grammatical gender rules were not sufficient to identify grammatical gender for the entire data set. A substantial number of nouns remained unassigned. Hence original data set was reduced to create a new data set which held only those entries which cannot be assigned a grammatical gender using a rule. This new data set created is referred to as *Reduced Data Set* (RDS) The best model for *Reduced Data Set* (RDS) was picked by training various machine learning classifiers on reduced data set.

4.3 Stage Two: Classification

The reduced data set had 635 entries for which gender could not be determined using association rule mining. To decide which classification model will best represent our data, various classification algorithms available in WEKA 3.6.11 were executed. Table 4 gives the best performing classifications algorithm. Table 4 indicates that word and morphological features used together provide a better classification model with best F-score of 0.723[12] obtained by Multi-layer Perceptron. This neural network function based classifier works well with both categorical and continuous attributes, performs well on the test set and hence is chosen as a learning model for training.

Table 4. Classification Algorithm Results.

Features	Algorithm	Precision	Recal	F-Score
	IB1	0.768	0.758	0.692
Word	LogitBoost	0.528	0.726	0.611
	MultilayerPerceptron	0.657	0.705	0.615
	Random Forest	0.612	0.747	0.63
	IB1	0.598	0.579	0.588
Morphological	LogitBoost	0.698	0.695	0.697
	MultilayerPerceptron	0.589	0.579	0.584
	Random Forest	0.669	0.684	0.676
	IB1	0.705	0.705	**0.705**
Word+Morphological	LogitBoost	0.528	0.726	0.611
	MultilayerPerceptron	0.734	0.716	**0.723**
	Random Forest	0.72	0.747	**0.712**

4.4 Stage Three: Staged Approach

The rules obtained in stage one and training model namely multilayer perceptron obtained in stage two were combined into a staged approach. The pseudo-code of the algorithm used by the staged approach to assign grammatical gender to noun entries in test data set is illustrated in Figure 1.

The experimental results obtained and evaluation are presented in the next section.

5 Experimental Results and Evaluation

The goal of the experiment was to build a grammatical gender identification system. Two approaches namely machine learning classifiers approach and staged

[12] Here Precision, Recall and F-score are the weighted average values generated

Algorithm 1: Assign Grammatical Gender to Nouns in WordNet

```
for each noun in WordNet
   if noun compatible with grammatical gender rule GR
       gender = GR(noun);
   else
       gender = MultilayerPerceptronClassifier(noun);
   end if
end for
```

Fig. 1. Algorithm for assigning grammatical gender to nouns in WordNet.

approach were used for the same. A separate test data set with 200 entries was created to test the performance of the approaches. The staged approach algorithm presented in Figure 1 was tested on the test data set to determine its performance. Table 5 tabulates the results obtained using staged approach. Table 6 compares the staged grammatical gender identification approach with best performing machine learning classifier. A minor improvement is seen in the weighted average values of precision, recall and F-score in the staged approach. The results obtained were analyzed and are presented in the next subsection.

Table 5. Grammatical Gender Identification using Staged Approach.

Grammatical Gender	**True Positives**	**False Positives**	**False Negatives**	**Precision**	**Recall**	**F-Score**
Feminine	91	5	4	0.95	0.96	0.95
Masculine	50	11	15	0.82	0.77	0.79
Neuter	31	12	7	0.72	0.82	0.77
Non-neuter	0	0	2	0	0	0

Table 6. Staged Approach Vs Machine Learning Classifiers.

Algorithm	**Precision**	**Recall**	**F-Score**
Staged Approach	0.854	0.86	**0.84**
Random Sub Space	0.78	0.807	0.79
Logistic Model Tree	0.807	0.837	0.82

5.1 Analysis of Results

To find out where the algorithm fails, the results obtained were analyzed from two perspectives namely association rule mining and classification model.

Association Rule Mining Analysis: Analysis of grammatical gender identification rules generated by association rule mining indicated that the feminine gender is identified using these rules . As expected the precision and recall for the feminine gender is high as can be seen in Table 5. An example where assignment of feminine gender fails is for the Konkani word "*shaeta*" (goldsmith) which has grammatical gender masculine but is assigned as feminine. This is an exceptional case wherein its morphological features resemble those of a feminine noun. Another example of failure is word like *baenagalaii* (Bengali language/Bengali person) which has multiple senses wherein the different senses have different gender namely in the Bengali language sense the noun is feminine and in the Bengali person sense the noun is masculine. The algorithm fails for such cases. A few instances of masculine and neuter gender can be identified using rules, but most of masculine and few neuter gender nouns cannot be identified based on rules. There are no rules generated corresponding to non-neuter gender.

Classification Model: The classification model has been used mainly to identify masculine, neuter and non-neuter genders. This word and morphological features used can classify nouns in the above stated gender values with a best F-Score of 0.72. Additional features like context based features may be used to improve the performance of such classifiers.

6 Conclusion

Grammatical gender of a noun in Konkani can be determined based on morphological and word features using a staged approach, with a weighted average F-Score of 0.84 thus supporting our hypothesis. On examination of grammatical gender identification rules we found that some morphological paradigms suggest grammatical gender. For some words like lawyer two paradigms get assigned suggesting that the noun has non-neuter grammatical gender. Some association rules example $BeginVowel = NULL\ EndVowel = o \rightarrow PID = twozero\ Gender = masculine$ acc:(0.99315) generated can also be used to automatically assign morphological paradigms to new noun entries based on word features. Future work will be focused on finding word, morphological or context features which give more clues to determine masculine and neuter gender.

Acknowledgments

We wish to express our gratitude to Prof. Kiran Budkuley, Dean Faculty of Language and Litterature, Goa University for her encouragement and all the support provided to clarify some of the doubts with respect to Konkani Language. We also acknowledge that we were able to carry out the work using the Konkani WordNet which was developed as part of the Indradhanush WordNet Project funded by Department of electronics and information technologY (DeitY), New Delhi 11(13)/2010-HCC(TDIL) dated 03/08/2010

References

1. Bhattacharyya, P.: Indowordnet. In: Proceedings of Seventh International Conference on Language Resources and Evaluation (LREC), Malta (2010) 3785–3792
2. Nicholson, J., Nordlinger, R., Baldwin, T.: Deep lexical acquisition of type properties in low-resource languages: A case study in wambaya. In: Proceedings of the 26th Pacific Asia Conference on Language,Information and Computation. (2012) 75–81
3. Babych, B., Geiger, J., Rosell, M., Eberle, K.: Deriving de/het gender classification for dutch nouns for rule-based mt generation tasks. In: Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra) @ EACL 2014, Gothenburg, Sweden (2014) 75–81
4. Dinu, L., Niculae, V., Sulea, M.: Dealing with the grey sheep of the romanian gender system, the neuter. In: Proceedings of COLING 2012: Demonstration Papers, Mumbai, India (2012) 119–124
5. Nastase, V., Popescu, M.: Whats in a name? in some languages, grammatical gender. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore (2009) 1368–1377
6. Spalek, K., Franck, J., Schriefers, H., Frauenfelder, U.: Phonological regularities and grammatical gender retrieval in spoken word recognition and word production. Journal of Psycholinguistic Research **37** (2008) 419–442
7. Vigliocco, G., Vinson, D., Paganelli, F.: Grammatical gender and meaning. In: Proceedings of the 26thMeeting of the Cognitive Science Society. (2004)
8. Cucerzan, S., Yarowsky, D.: Minimally supervised induction of grammatical gender. In: Proceedings of HLT-NAACL, Edmonton (2003) 40–47
9. Kennedy, C., Boguraev, B.: Anaphora for everyone: Pronominal anaphora resolution without a parser. In: Proceedings of COLING, Boulder,Colorado (1996) 113–118
10. Bergsma, S., Lin, D., Goebel, R.: Glen, glenda, or glendale: Unsupervised and semi-supervised learning of english noun gender. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), Boulder,Colorado (2009) 120–128
11. : Noun class (2015) ”http://en.wikipedia.org/wiki/Noun_class”.
12. : Grammatical gender (2015) ”http://en.wikipedia.org/wiki/Grammatical_gender”.
13. Desai, S., Desai, N., Pawar, J., Bhattacharya, P.: Autoparse: An automatic paradigm selector for nouns in konkani. In: Proceedings of the 11th International Conference on Natural Language Processing (ICON), Goa,India (2014) 154–162
14. Borkar, S. In: Konkani Vyakran. Konkani Bhasha Mandal (1992)
15. Almeida, M. In: A Description of Konkani. Thomas Stephens Konknni Kendr (1989)
16. Sardessai, M. In: Some Aspects of Konkani Grammar. Unpublished M.Phil. Thesis (1986)

Reviewing Committee of the Volume

Ajith Abraham
Bayan Abushawar
Hanady Ahmed
Hend Alkhalifa
Rania Al-Sabbagh
Marianna Apidianaki
Mohammed Attia
Aladdin Ayesh
Alexandra Balahur
Sivaji Bandyopadhyay
Srinivas Bangalore
Leslie Barrett
Roberto Basili
Nuria Bel
Anja Belz
Pushpak Bhattacharyya
António Branco
Nicoletta Calzolari
Nick Campbell
Michael Carl
Violetta Cavalli-Sforza
Niladri Chatterjee
Khalid Choukri
Kenneth Church
Dan Cristea
Walter Daelemans
Samhaa El-Beltagy
Michael Elhadad
Ossama Emam
Aly Fahmy
Anna Feldman
Alexander Gelbukh
Dafydd Gibbon
Gregory Grefenstette
Ahmed Guessoum
Nizar Habash
Kais Haddar
Lamia Hadrich Belguith
Eva Hajicova
Sanda Harabagiu
Yasunari Harada
Karin Harbusch
Ales Horak
Veronique Hoste
Nancy Ide
Diana Inkpen
Aminul Islam
Sattar Izwaini
Guillaume Jacquet
Doug Jones
Sylvain Kahane
Alma Kharrat
Adam Kilgarriff
Philipp Koehn
Olga Kolesnikova
Valia Kordoni
Leila Kosseim
Mathieu Lafourcade
Mark Lee
Krister Lindén
Bing Liu
Elena Lloret
Bente Maegaard
Sherif Mahdy Abdou
Cerstin Mahlow
Suresh Manandhar
Sun Maosong
Diana Mccarthy
Alexander Mehler
Farid Meziane
Rada Mihalcea
Evangelos Milios
Jean-Luc Minel
Dunja Mladenic
Marie-Francine Moens
Herman Moisl
Masaki Murata
Preslav Nakov
Roberto Navigli
Joakim Nivre
Kjetil Nørvåg
Attila Novák
Kemal Oflazer
Constantin Orasan
Ekaterina Ovchinnikova
Ivandre Paraboni
Saint-Dizier Patrick
Maria Teresa Pazienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Stelios Piperidis
Octavian Popescu
Soujanya Poria
Marta R. Costa-Jussà
Ahmed Rafea
Allan Ramsay
Mohsen Rashwan
German Rigau
Fabio Rinaldi
Horacio Rodriguez
Paolo Rosso
Vasile Rus
Horacio Saggion
Franco Salvetti
Rajeev Sangal
Kepa Sarasola
Roser Sauri
Hassan Sawaf
Satoshi Sekine
Nasredine Semmar
Khaled Shaalan
Bernadette Sharp
Grigori Sidorov
Vivek Kumar Singh
Vaclav Snasel
Efstathios Stamatatos
Josef Steinberger
Jun Suzuki
Stan Szpakowicz
William Teahan
J.-M. Torres-Moreno
George Tsatsaronis
Dan Tufis
Olga Uryupina
Renata Vieira
Manuel Vilares Ferro
Aline Villavicencio
Piotr W. Fuglewicz
Bonnie Webber
Savas Yildirim
Imed Zitouni

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2015
Printing 500 / Edición 500 ejemplares