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 70,** noviembre 2013. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121511550100-102, 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 70**, November 2013. 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 70
Volumen 70

Advances in Computational Linguistics

Alexander Gelbukh (Ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2013

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

(Prefacio)

This volume of "Research in Computing Science" presents a selection of papers on computational linguistics by authors from 14 countries: Canada, China, Egypt, Finland, France (6), Germany (2), India (2), Japan, Mexico (3), New Zealand, Romania, USA, Spain, and Vietnam. The papers have been carefully chosen based on reviews by the members of the international reviewing committee of this volume.

Computational linguistics, also referred to as natural language processing or human language technologies, is a branch of science in the intersection of linguistics and artificial intelligence that has two main objects of study. On the one hand, it studies the structure of human languages using computational tools, which give to the linguist a lot more power than traditional paper-and-pencil techniques. Computers are capable of processing huge amounts of text to suggest to the linguist hypotheses about possible regularities in the language or to test the linguist's own hypotheses about the structure and statistical properties of the human language. In addition, computers can easily adapt to different genres and thematic domains of texts.

On the other hand, natural language processing has the goal of enabling computers to accomplish meaningful tasks connected with text and speech by using linguistic knowledge, both encoded by human linguists in the form of grammars and dictionaries and learnt automatically. Modern machine learning methods are capable of automatically learning linguistic regularities from huge text corpora and using this information in order to accomplish the required natural language processing tasks.

In this volume, twenty papers present a wide range of problems and solutions in the area of computational linguistics and natural language processing. The first paper uses computational methods to explore an ancient writing system. The next papers evaluate word segmentation techniques in Chinese, propose treatment of named entities, suggest new approaches to tagging in languages of India, to measuring semantic relatedness, to word sense disambiguation, to detecting person's names in Arabic, to solving ambiguities in discourse parsing and parsing of intentions, and to recognizing textual entailment.

The next group of papers is devoted to various aspects of lexicography, such as extracting parallel sentences from Wikipedia, improving biomedical term extraction, exploring various types of approaches to multilingual terminology extraction, and learning for relation extraction. Finally, several papers are devoted to various aspects of analysis of text, such as mining for suggestions and outdating information, analyzing changes in quotes over time, genre identification, building a collocation dictionary, and evaluation of existing approaches to text summarization.

Alexander Gelbukh November 2013

Table of Contents

(Índice)

Page/Pág.

Comparison between Rongorongo and the syllable sequence of ancient chants from the Easter Island

Fumihiko Yamaguchi

Keio University
Department of Information and Computer Science
Hiyoshi, Kanagawa, 3-14-1, JAPAN
yamagu@nak.ics.keio.ac.jp

Abstract. Rongorongo is sequence of symbols engraved on the surface of wooden artifacts made in the Easter Island. Rongorongo is considered to be script, but has not yet been deciphered. As Rongorongo is assumed to be the record of chant, we syntactically compare Rongorongo with chants in the Easter Island. A simple exhaustive search method is applied to find correspondence between the order of symbols in Rongorongo and that of syllables in the chants. However, the correct correspondence is unknown. Therefore, the proposed method is adapted to a similar problem of matching kanji-hiragana mixed text with katakana text in order to characterize the method. After confirming that the method produces higher precision when there are more kind of frequent symbols. As a result, some correspondences are found, whereas no correspondence is found in most pairs of Rongorongo lines and verses of the chants.

1 Introduction

In the Easter Island, sequence of symbols engraved in wooden artifacts are remained which is called Rongorongo. Rongorongo is considered that may be scripts, however it has not yet been deciphered.

The first European visitors to the Easter Island is Roggeveen, who is a Dutch explorer, arrived on Easter Day 1722. But they made no mention of written characters. The second European is Don Felipe arrived in 1770. There is a record that Easter Island people signed on the Spanish document with local symbols. However the first mention of the wooden artifacts covered with symbols is by Eyraud in 1864. Bishop Jaussen, who had been sent to Tahiti in the 19th century, owned some Rongorongo artifacts. He met a man named Metoro who is a native person of the Easter Island. As Jaussen let Metoro read the artifacts, Metoro read it in a singing voice. However, Rongorongo remained undeciphered because Metoro did not know what he said[6]. Emory surmised that Rongorongo was invented after that Easter Island people looked the Spanish document and for recording their chants[5].

It is remarkable that Metoro's reading was like chanting, though it was not significant to the content of his reading. Because even if he was actually not

Fig. 1. Rongorongo engraved on Mamari, side A, line 2.

able to read, there is a possibility of mimicking an original appearance in which Rongorongo is read. If Rongorongo is a document which record chants, some relations might be seen between the sequence of Rongorongo and the sequence of phoneme of the chant. Jorge Hotus Salinas collected the old chants in the Easter Island, and recorded them in Latin character[10].

If Rongorongo symbols are characters, each symbol may have its pronunciation which is a phonetic sequence. If a sequence of Rongorongo symbol represents a part of a verse of a chant, it is thought that the symbols appear in the sequence in the same order as their pronunciation of the verse. In this paper, only syntactic relation is considered, without considering the meaning of the Rongorongo symbol nor meaning of the word in the verse. When each of Rongorongo and pronunciation of verse is expressed by distinct sequences, the problem is to find a relation between two sequences. In this paper, lines of Rongorongo are expressed in sequences of codes called Barthel code, and the chants are expressed in sequences of syllables. However, even if there is a set of pairs (a, b) of symbols in sequences A and B such that a appears in A in the same order of b appearing in B, it's not always means that A corresponds to B. For example, assume $A = xy$ and $B = 1234$, there are many correspondence such as $\{(x, 1), (y, 2)\}$, $\{(x, 2), (y, 4)\}$ and so on. Its easy to see that both sequence needs some length. And a symbol which uniquely appears in one sequence can be paired to any symbol in the other sequence. Therefore, we should consider symbols which appears twice or more in each sequence. In this paper, an algorithm is proposed which finds a correspondence whether two sequence have a set of pairs of frequent symbols which appears in the same order in each sequence. Our interest is to apply the algorithm to Rongorongo and the chants, but we cannot evaluate any result directly, because Rongorongo is undeciphered. Thus, we apply the algorithm to known language. In this paper, Japanese texts are employed to characterize our method. And then we will show the output of the algorithm applied to Rongorongo.

Some natural language processing methods are expected to have little dependency on language specific features. There are some NLP researches about correspondence between a lost language and phonetic sequences, while lost languages are gaining increasing interest in NLP community [8]. Snyder et al investigated computer aided decipherment of lost language[12]. They extract pairs of Ugarit word and its cognate in Hebrew which is known language close to Ugarit geographically and temporally. Knight et al. investigated HMM-base decipherment of character substitution cipher which relates an unknown script that represents a known spoken language [7]. Their method find character-to-sound mappings from non-parallel character and sound sequences. Sidorov et al. investigated

searching tool for Mayan character [11]. They pointed out the importance of character identification, original scripts are the image data but computer readable texts.

Since remained Rongorongo is small and sparse corpus, we employ exhausitive search rather than statistical methods.

2 Rongorongo and Chants

Each Rongorongo artifact is named after local words or the city of which the possessing museum addressed. Most artifacts are tablets and its side is distinguished by side A/B or Verso/Recto. Rongorongo is written in boustrophedon style. Moreover the glyphs are upside down in each other line.

Barthel classified each symbol of Rongorongo in its shape and encoded them by three digits [2, 3]. This is called Barthel code. Barthel coding classify Rongorongo symbols in more detail than Mètraux's classification. By using Barthel code, Rongorongo can be treated as a computer readable text information. Yamaguchi et al. developed the tool that supports to encode from Rongorongo image data to sequences of Barthel codes [13]. In this paper, the sequences of Barthel code are obtained using this tool. The image of the Rongorongo symbol originally downloaded from rongorongo.org[1]. In this paper, the four major artifacts are examined that are named Aruku Kurenga, Tahua, Mamari and Keiti respectively. All these names come from local words. These artifacts were once owned by Jaussen[6]. They includes 83 lines total. There are 441 kinds of symbol counted in Barthel code, each line includes 60.5 Barthel codes in average.

The Salinas' archive of Easter Island chants includes 89 titles. Most titles include 3 or 5 variant of verses. Some chants also have another part of chorus. There are totally 372 verses.

Rongorongo symbol seem to be in the natural shape such as grass, fish and birds. The example of Rongorongo line is shown in Figure 1. Mètraux classified the Rongorongo symbols into about 120 kinds. He also pointed out that there are too few kinds of symbols to think of Rongorongo as ideographic script and too much as phonographic[9].

Therefore, Rongorongo is assumed to be syllabic or logo-syllabic in this paper, and thus each verse of the chant is treated as a sequence of syllables. According to Bellwood's family tree for the Fijian and Polynesian languages, Tahitian and Easter Island are closely related[4], such that both has the common ancestor Proto-Central Polynesian language. Moreover, the modern spoken language of the Easter Island is influenced by Tahitian language. A syllable of Tahitian language consists of one vowel or a consonant and a following vowel. And 'n' and 'm' can be appeared solitary. So, in this paper, each of them is considered as a syllable by itself. Each verse of the chant is successfully divided into syllables in the same way. There are few exception that solitary 'k' in a word 'kraverita' and solitary 's' in a word 'Mas'. However, each of these exceptions occurred only once in a verse. Thus, as descrived below, the exception does not influence to our

```
corres(i_R, i_C, H) :=
 1: if i_R ≥ length_R then return true endif;
 2: if i_C ≥ length_C then return false endif;
 3: if the frequency of R[i_R] ≤ 1 then return corres(i_R + 1, i_C, H) endif;
 4: if the frequency of C[i_C] ≤ 1 then return corres(i_R, i_C + 1, H) endif;
 5: if there exists X such that (R[i_R], X) ∈ H then
 6:   if X = C[i_C] then
 7:     if Rest_R[i_R] > Rest_C[i_C] then return false endif;
 8:     if corres(i_R + 1, i_C + 1, H) then return true endif
 9:   endif;
10: else
11:   if Rest_R[i_R] ≤ Rest_C[i_C] then
12:     if corres(i_R + 1, i_C + 1, H ∪ {(R[i_R], C[i_C])}) then return true endif
13:   endif;
14: endif;
15: return corres(i_R, i_C + 1, H).
```

Fig. 2. Psudo code for judging correspondence between a Rongorongo line and a chant

experiment. There are 119 kinds of syllables used in the chant and each verse includes 35.0 syllables in average.

3 Correspondence

When a whole line of Rongorongo (sequence of symbols) represents a part of a verse (sequence of syllables), there exists a set of pairs of symbol and syllable such that they occur in the respective sequences in the same order. This relation is examined by exhausitive search. The algorithm is shown in Figure 2. In this code, R and C are the sequence which are intended a line of Rongorongo and the sequence of syllables of a verse of a chant respectively. The arguments of function `corres` are the index i_R and i_C of the sequences and hypothesis H of correspondence. $R[i_R]$ represents the i_R-th symbol of R, $length_R$ is the length of R, $Rest_R[i_R]$ is the number of frequent symbols in R after i_R-th symbol and so is in C. H is a set of pairs of symbols from R and C. The *Rest* are used to cut search time. The call of `corres`$(0, 0, \emptyset)$ returns false when there is not possible correspondence. Otherwise, it returns true.

When R contains n kinds of frequent symbols, and C contains m frequent syllables, the algorithm checks two cases for each occurrence of frequent symbols whether it corresponds to the current frequent syllable or not. The pseudo code runs in $O(m2^n)$ in time.

As Rongorongo is unknown to be truly syllabic, it is assumed that the pronunciation of a symbol consists of zero or more syllables. Even if Rongorongo is ideogram, one symbol might corresponds to a sequence of syllables in some length (that is the pronunciation of the symbol) and thus the order of symbols

and the order of syllables will be allinged. However, when all association is enumerated, combinatorially many association will be found. The pseudo-code is to judge whether there is possible correspondence or not. Therefore, the algorithm does not enumerate all these association. Instead, it returns true when one association is found.

The correspondence of symbol and syllable is not exclusive: different symbols may correspond to the same syllable. Since unique symbols and its corresponding syllables do not constrain the check of the same order occurrence, only symbols and syllables which occur twice or more in the respective sequences are in concern. They are called frequent symbols and frequent syllables respectively. A frequent syllable corresponds to a symbol is thought as a part of the pronunciation of the symbol.

3.1 Experiments in known language

Our main purpose is to extract correspondence between lines of Rongorongo and verses of chants. However, because Rongorongo is undeciphered, we cannot evaluate the result directly. Therefore, we apply the proposed algorithm to known language in order to characterize the method. In this subsection, we describe about experiment on Japanese texts.

Japanese writing system have three distinct kinds of symbols, hiragana, katakana and kanji. Both hiragana and katakana are syllabaries and they have one-to-one correspondence, i.e. each syllable has two syllabic symbols. Kanji is logogram and each kanji character has some pronunciations which are able to be written in a sequence of one or more characters in katakana or hiragana. In usual modern Japanese writing, kanji and hiragana are used mixedly and katakana is used for loanwords etc.

For an experiment of a known language case, a set J of 20 kanji-hiragana mixed sentences is prepared each of which is extracted randomly from a Japanese math book. Each sentence $j \in J$ is in length of 21-41 characters. And a set K of 20 katakana sentence is prepared so that for each $j_i \in J$, $k_i \in K$ is the same pronunciation as j_i. Note that there is no intersection between the sets of characters in J and the set of characters in K.

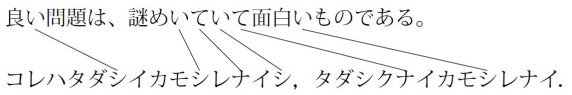


Fig. 3. An example of misjudged case

The algorithm is applied to each of $20 \times 20 = 400$ pairs of all combination, to judge whether there is possible correspondence in which frequent symbols appears in the same order. The test set of 400 pairs contains 20 correct pairs

Table 1. Frequent hiragana and katakana correspondence in Figure 3

hiragana	katakana
い	シ
て	ナ

which are of kanji-hiragana mixed text and its pronunciation in katakana. As a result, for all of the 20 correct pairs, the algorithm returns true. And it returns false for 211 pairs of the 380 incorrect pairs. However, there are 169 patterns are misjudged. Those pairs are not correct pairs, nevertheless, there exists pairs of frequent symbols appearing in the same order in each sequence. An example of misjudged case is shown in Figure 3 and the correspondence between characters of the figure is shown in Table 1. In this example, there are only 2 frequent symbols in kanji-hiragana text, where katakana text contains 8 frequent symbols.

This is a typical characterization of many misjudged cases that the number of frequent symbols in j is inclinable less than that of correct case.

Table 2. The number of frequent symbols in kanji-hiragana sentence and the number of misjudged combination

n	# of misjudge	recall(%)	precision(%)	F-measure
2	74	100	11.9	0.213
3	67	100	13.0	0.230
4	56	100	15.2	0.263
5	23	100	30.3	0.465
6	16	100	38.5	0.556
7	9	100	52.6	0.690
8	6	100	62.5	0.769

Therefore, for another experiment, the sets J_n of 10 kanji-hiragana mixed sentences are prepared, for each $2 \leq n \leq 8$ where n indicates the number of frequent characters in kanji-hiragana mixed sentence. The algorithm is applied to all of 100 pairs for each n which contains 10 correct pairs. The result is shown in Table 2. For every n , the algorithm returns true for all correct pairs (thus recall is 100%). And the more n is, the less pairs are misjudged.

As a discussion, we found a rare case of misjudge for correct pairs. Because a kanji may have two or more pronunciation, there can be a case that a kanji is the frequent symbol in j and its different pronunciations are contained in the corresponding k. In this case, the proposed algorithm misjudged for this correct pair.

3.2 Experiment in Rongorongo and Chants

All of the 30,876 combination, that is 83 Rongorongo lines × 372 verses, are examined by the algorithm. As a result, 618 pairs are judged that there are possible correspondence. The rest 30,258 pairs are confirmed that they don't relate. As the proposed method is exhausitive search, when there is possible correspondence between frequent Rongorongo symbols and frequent syllables, they are expected to be found.

Table 3. Rongorongo and verse of chant which may have correspondence

Aruku Kurenga, Verso, line 1	
1. E Ira e, e Raparenga e #5	17. Ka u manu a roto #1
2. E Renga Mariki #3	18. Ko Tongariki #2
3. E mea tino mamahi rua e #4	19. Ko Tongariki #3
4. E nui te tamu #2	20. Kraverita #1
5. E nui te tamu #3	21. Kraverita #1 coro
6. I he a Hotu Matu'a e hura nei #4	22. Mai runga #3
7. Ka e'a te neru #2	23. Marìa Reina #1 coro
8. Ka haro au i vai a repa #1	24. Marìa Reina #2
9. Ka haro au i vai a repa #2	25. O mea o te hare #1
10. Ka memea(1) #1 coro	26. O mea o te hare #3
11. Ka memea(1) #3	27. Poio nuinui a Tuki #3
12. Ka memea(2) #1 coro	28. Renga mitimiti #3
13. Ka memea(2) #3	29. Renga varevare #4
14. Ka memea(2) #4	30. Tamaiti puai rahi #3
15. Ka tea mai te niho #4	31. Te pito #1
16. Ka tere te vaka(2) coro	32. Tuki horo pari #2
Mamari, side A, line 2	
1. E Ira e, e Raparenga e #5	4. Ka tea mai te niho #4
2. E nui te tamu #3	5. Ko Tongariki #2
3. Ka e'a te neru #2	
Mamari, side B, line 2	
1. Ka e'a te neru #2	2. Ka tea mai te niho #4

However, there are possibilities that the algorithm returns true even when the Rongorongo line is not the representation of the verse. As we shown by the Japanese case experiment, the number of frequent symbols will work as a good filter. Therefore, Rongorongo lines with many frequent symbols are remarkable. In each of 69 lines of Rongorongo out of 83 lines, there are 8 frequent symbols. There are only 39 pairs out of 618 pairs when the Rongorongo lines are selected to this 69 lines. These relatively significant pairs are shown in Table 3. In other words, Rongorongo lines of 579 pairs, in which the algorithm finds possible correspondence, contains 7 or less frequent symbols.

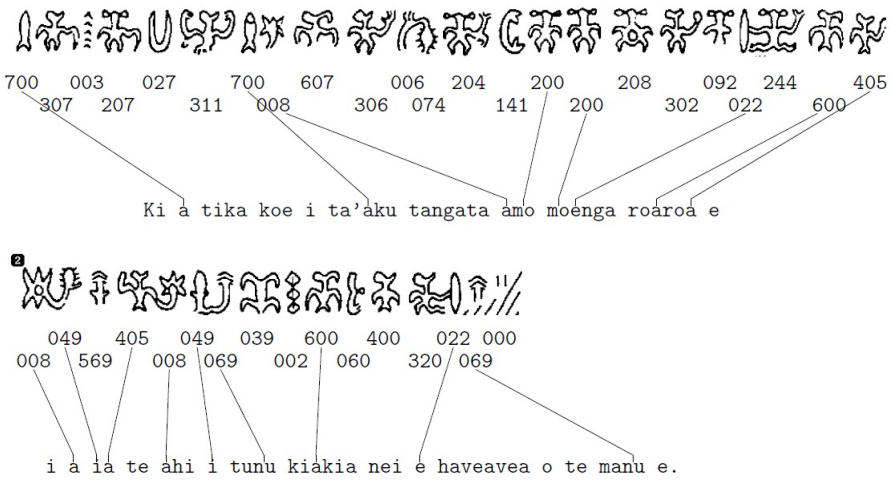


Fig. 4. Example of correspondence, Aruku Kurenga Verso 1 and "E Ira e, e Raparenga e" #5

Table 4. Symbol and syllable correspondence in Figure 4

Code	Syllable	Code	Syllable
700	a	008	a
200	mo	022	e
600	a	405	a
049	i	069	nu

In Table 3, list of possibly corresponding verse is shown for each Rongorongo line. Each verse is represented by its title and the variation. For example, "E Ira e, e Raparenga e #5" represents the 5th verse of chant titled "E Ira e, e Raparenga e". As is shown in this table, the result is not exclusive. There are 32 verses corresponding to Aruku Kurenga Verso line 1, and some verses such as "Ka tea mai te niho #4" correspond to 3 lines of Rongorongo.

The correspondences between each frequent symbol and each frequent syllable are extracted. An example of found correspondence is shown in Figure 4. In this figure,there are the image of Rongorongo (the first line of Aruku Kurenga, Verso), the sequence of Barthel codes and the 5th verse of chant titled "E Ira e, Raparenga e". Each straight line connecting a Barthel code and the verse

represents correspondence between symbols and syllables. The correspondence is shown in Table 4, which contains Barthel code, typical Rongorongo image of the code and corresponding syllable. As the judge algorithm returns when one correspondence is found, there may be another correspondence between each frequent symbols and syllables. The 2nd line of Mamari side-A which corresponds with the second most number of verses is shown above in the Figure 1.

4 Conclusion

It is examined whether the order of the codes in lines of Rongorongo corresponds to the order of the syllables in the chants in the Easter Island. As a result, 618 combination of correspondence are found. However, when Rongorongo contains few kinds of frequent symbols, the correspondence can be found accidentally. By selecting Rongorongo lines which contains many kinds of frequent symbols, relatively plausible 39 correspondences between lines of Rongorongo and verse of chant are listed. However, it might be a rashness to think the direct relation to the decipherment because the correspondence is not exclusive.

We cannot say that Rongorongo represents the verse even if the order of the symbols corresponds to the order of syllables. However, the Rongorongo line does not represent the verse when the orders do not meet, under the assumption that Rongorongo is syllabic or logo-syllabic and the whole line of Rongorongo represents the part of verse. The main contribution of this paper is that it is confirmed by a exhausitive search that the 30,258 combinations of Rongorongo line and verse do not relate, while there are few record or other corpus of old language in the Easter Island.

For the pairs of Rongorongo line and verses which is judged positively in this study, there are combinatorially many correspondence of symbol and syllable under the condition that the same symbol cannot corresponds to different syllable. As the future work, extracting plausible hypothesis about reading (or part of syllables) representing some symbols will be considered. There might be clues such as the distances between related symbols and syllables in the respect sequences. Once pronunciation of some Rongorongo symbols are assumed, the meaning will be inferred by comparing modern local language of the Easter Island.

Acknowledgment

This work was supported by MEXT KAKENHI Grant Number 12877681.

References

1. Rongorongo or the hieroglyphs of the easter island tablets (accessed in Oct. 2003). retrieved from `http://www.rongorongo.org/`.

2. Barthel, T. S. (1958a). *Grundlagen zur Entzifferung der Osterinselschrift*, volume 36 of Abhandlungen aus dem Gebeit der Auslandskunde 64, Reihe B. Hamburg: Cram, de Gruyer & Co.
3. Barthel, T. S. (1958b). *Talking Boards of Easter Island*, Scientific American 198. pages 61–68.
4. Bellwood, P. (1987). *The Polynesians: Prehistory of an Island People.* Revised edition. London: Thames & Hudson.
5. Emory, K. P. (1972). *Easter Island's position in the prehistory of Polynesia.* Journal of Polynesian Society, vol. 81. pages 57-69.
6. Fischer, S. R. (1997). *RONGORONGO*, volume 14 of *Oxford Studies in Anthropological Linguistics.* Clarendon Press, Oxford.
7. Knight, K., Nair, A., Rathod, N. and Yamada, K. (2006) *Unsupervised analysis for decipherment problems*, In the Proceedings of the COLING/ACL, pp. 499–506, 2006
8. Knight, K. and Sproat, R. (2009). *Writing systems, transliteration and decipherment*, NAACL Tutorial, 2009
9. Mètraux, A. (1940). *Ethnology of Easter Island*, volume 160 of *Bernice P. Bishop Museum Bulletin.* Bishop Museum Press, Honolulu.
10. Salinas, J. H. Cantos antiguos de rapa nui. Archive of Museo Antropologico, Chile, Easter Island, (the year of production is unknown).
11. Sidorov, G., Pichardo-Lagunas, O., and Chanona-Hernandez, L. (2009). Search Interface to a Mayan Glyph Database based on Visual Characteristics, Lecture Notes in Computer Science, Vol. 5723, Springer-Verlag, pp. 222–29, 2009.
12. Snyder, B., Barzilay, R., and Knight, K. (2010). A statistical model for lost language decipherment. In *proc. ACL 2010.*
13. Yamaguchi, F., Nobesawa, S., and Sato, K. (2003). Rongorongo character listing tool. In *Proceedings of the 3rd IASTED International Conference on Visualization, Imaging, and Image Processing*, volume 1, pages 158–162, Benalmadena, Spain.

Distributional Evidence and Beyond: The Success and Limitations of Machine Learning in Chinese Word Segmentation

Jianqiang Ma, Dale Gerdemann

Department of Linguistics, University of Tübingen
Wilhelmstr. 19, Tübingen, 72074, Germany
jma@sfs.uni-tuebingen.de, dg@sfs.uni-tuebingen.de

Abstract

In this paper, we argue that the key to the success of the current state-of-art statistical learning algorithms for Chinese word segmentation (CWS) mostly lies in their optimal weighting of non-overlapping *distributional evidence* in the corpora. The utilization of distributional evidence is more essential than the learning algorithm. We further analyze the characteristics of distributional evidence for CWS, under the framework of Zipf's law and summarize the limitation of statistical learning in CWS as the *feature absence problem*, which may be apparent yet usually neglected. Making a connection between theoretical/empirical linguistics and CWS, we suggest that the study and development of a *generative word formation system* may be beneficial for both the science and engineering of CWS. We wrap up the discussion after reviewing some recent works that are already on this line.

Introduction

Tokens in general, words are considered as building blocks of linguistic structures of human languages and basic inputs for natural language processing (Webster and Kit 1992). In many Asian languages, including Chinese, sentences are written as character sequences without explicit word delimiters, thus tokenization or *word segmentation* remains a key research topic in language processing for these languages.

The most popular model among modern word segmenters is probably character position tagging (Xue, 2003), which views word segmentation as labeling the positional roles that character plays within words, using labels such as **B**eginning, **M**iddle, **E**nding and **S**ingleton. Under such formulation, Chinese word segmentation (CWS) becomes a special case of sequence labeling problem, which can be effectively solved by machine learning techniques such as conditional random fields (Lafferty et al., 2001), which achieves state-of-art results for CWS.

In recent years, the performance of machine learning based segmenters has been further pushed forward by model combination (Wang et al., 2010; Sun, 2010), utilizing unsupervised segmentation on unlabelled data (Zhao and Kit, 2008; Sun and Jia, 2011), jointly learning segmentation and POS tagging (Jiang et al., 2008; Zhang

and Clark, 2008; Sun, 2011), etc. On the other hand, it appears that out-of-vocabulary words (OOV) remain a major challenge even for these sophisticated systems. Given this background, our paper attempts to analyze both successes and limitations of machine learning approaches to CWS, in the hope of bringing new understandings and inspiring novel methods.

First of all, what types of *evidence* (information/feature) are most important for any segmenter? The most intuitive choice is *lexical forms*, which have been extensively used by early systems in the form of dictionary or token functions (e.g. frequency). However, as lexical forms are incapable of describing morphological behaviors of characters, it fails to contribute to recognition of OOV, which exist as a result of dynamic and productive word formation in Chinese. It turns out that character information alone provide adequate information for describing both IV (in-vocabulary words) and OOV, suggested by the success of various character position tagging systems. Specifically, such systems mostly rely on *character distributional evidence*, i.e. characters and character co-occurrences in different positions of words or word sequences.

Another important question is what role machine learning algorithms play. It might seem that the machine learning algorithm is a black box where magic happens, i.e. machine learning should get all the credit for the improvement over the well established baseline of maximum matching (Liang, 1986). But this needs more careful examination. We show in section 3 that the role of machine learning in CWS systems can be better described as *feature weight optimization*.

One implication of above mentioned issues is that despite different strategies for feature weight optimization, the performance of virtually *all* the current machine learning based segmenter are bounded by what can be expressed by character distributional evidence. Like many other linguistics phenomena, the character ngram distribution is characterized by *Zipf's law* (Zipf, 1949), which states that relatively few items are very frequent while most items are rare. Given Zipf's law, the distributional features that we have acquired from the training corpus are likely to cover only a subset of distributional features of the testing corpus, as some of rare features may only appear in either corpus but *not* both. This is consistent with our empirical study of distributional evidence and is exactly the problem for recognizing OOV. So the real challenge in CWS is that the distributional evidence for some characters in OOV is at least partly unavailable, where algorithmic predictions yield only low accurate guessing.

Similar to the limitation of machine learning in CWS, Yang (2011) suggests that usage/item-based theory in language acquisition (Tomasello, 2000; Hay and Baayen, 2005) has drawbacks on modeling the empirical data, also because of the Zipf's law. The generative linguistic system, on the other hand, is consistent with the language acquisition data. Interestingly, recent development of Chinese morphology, such as Packard (2000) and Xue (2001) also argues that it is attractive to describe word formation in Chinese using generative rules with part-of-speech like tags. These theoretical advancements shed light on new paths to solving the OOV problem in word segmentation with generative word formation models. Our discussion finishes

by summarizing some pilot work that are already in this direction, including work of the current authors.

Distributional Evidence for CWS

Early work in CWS extensively use lexical forms as the main information source. In maximum matching, sentence substrings that match lexical entries in the dictionary are selected as word candidates and the disambiguation of conflicting segmentations are achieved in a greedy search way. In finite state methods for CWS such as Sproat et al. (1996), lexicon is represented as weighted finite state machines and the segmentation disambiguation is based on scores of individual lexical item given by the finite state machine, which is mostly trained from word frequency statistics. But the rise of character position tagging approach to CWS shows that the lexical information is *neither necessary nor adequate* for the building accurate CWS systems. On one hand, various systems mainly using character distribution information (Xue, 2003; Peng et al., 2004) have similar results on IVs compared with word-based systems. On the other hand, character position tagging systems have very strong power on OOV recognition, which word-based systems basically fail to do.

Note that even for recent discriminative learning powered word-based segmenters (Zhang and Clark, 2008) that have state-of-art performances, character level features have been widely integrated. Actually, while it is hard to imagine how OOV can be properly modeled if all the character information is removed, discarding all lexical information may just end up with a system somewhat similar to a character tagging system.

1.1 Character Features that matter

In fact, lexical forms can be viewed as a special case of character distributional information, as the lexicon is a set of character sequences (co-occurrences). Some of the most useful character features proposed in Xue (2003) are following:

- Character unigrams: C_s (i-2<s<i+2)
- Character bigrams: $C_s C_{s+1}$ (i-2<s<i+2)
- Tag unigrams: T_s (s=i-1, i-2)

, where C represents a character, T represents a tag, s denotes the position index of the character string and i denotes the position of the current character of interest.

It can be seen that besides the interactions with character position tags, features are basically *character co-occurrences*. This feature set has been widely adopted in many latter systems, complemented by extra features such as punctuation, date, digit and letter, tone, etc. (Zhao et al., 2006). But it is fair to say the improvement brought by extra features is useful yet marginal.

1.2 The minority rules

It is not adequate to know that character distributional evidence is the dominant information for segmentation. As features do not necessarily contribute equally to the task, it is more interesting to examine *how* different features influence the

segmentation. Feature pruning provides a good perspective to understanding the contributions of individual features. If some features are pruned without significantly hurting the performance, these features may be less crucial or at least redundant with regard to the remaining features. Zhao & Kit (2009) have proposed a simple and efficient model pruning method for conditional random fields. A closer look at their experiments results on CRF based CWS helps us better understand the roles that different features play. The general message is that standard features as mentioned in previous section are *highly redundant*. According to their report, the model that uses only 2% of total number of features that have survived the pruning process can still reach above 97% of the accuracy of that which can be accomplished with the full feature set. Moreover, no performance loss occurs at all until the pruning rate is larger than 65%. In other words, a few features contribute a great deal to the performance of the current state-of-the art system.

Researchers have also found similar patterns on other sequence labeling tasks such as named entity recognition and chunking as well (Goldberg & Elhadad, 2009). It has shown that accurate models for these tasks can be learned from a heavily pruned feature space, which contains less than 1% of the features in the training set. In their experiments it turns out that rare features are used for ruling out uncertain cases by the machine learning algorithm rather than learning useful generalizations. We speculate that this conclusion might also be true for CWS task and we will further discuss the characteristics of the distribution of character ngram features in section 4.

The Role of Machine Learning

The Chinese language processing community has witnessed a dramatic performance boom of CWS systems since the introduction of machine learning algorithms under the character position tagging framework. It appears that machine learning is the black box where magic happens, as there is a huge gap between the state-of-the-art machine learning systems and the traditional dictionary-based greedy search baseline maximum matching. However, since most machine learning based systems dominantly rely on character distributional evidence, one may wonder whether the character distributional evidence within the framework of character position tagging should be given more credit that they have deserved. Our preliminary study has also shown that it is possible to achieve more than half of the error reduction on OOV recognition that the-state-of-art methods can achieve, by a simple combination of distributional evidence.

The re-examination of the role of machine learning in CWS is useful for a thorough understanding on *how* machine learning contributes to this task. To simplify the discussion, we restrict ourselves to log linear family of learning algorithms, i.e. maximum entropy, CRF, etc. These algorithms combine the features in a linear way and the learning process is reduced to the estimation of feature weights. But the argument should also hold for other algorithms, such as artificial neural networks, the only difference of which in this context is that there are hidden nodes that represents non-linear combinations of features. In either case, what machine learning *can* do is to optimize the weights for features using different strategies. Thus the role of machine

learning can be summarized as feature weight optimization. This understanding is important as one should distinguish the challenge in optimization for a given the feature space and the inherent problems of feature space itself. As we will show later sections, this links closely to the limitations of machine learning approaches to CWS and calls for new perspective of looking at CWS.

The Zipfian Distribution of Distributional Evidence

1.3 The feature absence problem of OOVs

OOVs are considered to be the major error source in the state-of-the-art machine learning based CWS systems. While those systems can achieve accuracy (F-score) over 95% on treebank corpora, their recall on OOVs are typically only around 70% (Emerson, 2005; Levow, 2006; Zhao and Liu, 2010). In order to illustrate the main problems of machine learning approaches to CWS, we have conducted an empirical study on those OOVs that the modern CWS systems fail to recognize. We are particularly interested in whether those errors are caused by feature weight optimization problems, or the inherent problems of the feature space itself.

The study is based on Penn Chinese Treebank version 5 (Xue et al., 2005), which is manually word-segmented. We trained a CRF based segmenter on 75% of the corpus and use the model to segment the remaining 25%. Those words only occur in the training section but not the testing section are considered as OOVs. The OOV rate is about 9% in this set-up. We use a simplified version of feature template proposed in Xue (2003) for training, namely only current characters (C_0), current and previous characters ($C_{-1}C_0$, denoted as B_1) as well as current and next characters (C_0C_1, denoted as B_2), i.e. unigrams and left/right bigrams. This choice is for the purpose of concentrating on the dominant factors and simplifying the discussion, given the fact that those features contribute more than 98% of the overall accuracy and 95% of OOV recall on this corpus.

One observation about those error-causing OOVs has drawn our attention. Among all character instances,

- 1.6% have C_0 feature unseen and thus B_1 and B_2 feature unseen in the training corpus (Type I);
- 29.2% have only C_0 feature seen, but both B_1 and B_2 features are unseen (Type II);
- 36.2% have and only have one of the bigram features unseen, i.e. either B_1 and B_2 is unseen (Type III).

In other words, 67% of character instances have at least one of the features B_1 and B_2 unseen from the training corpus, while only 23% of character instances have both B_1 and B_2 seen in the training corpus.

We may call this phenomenon as the *feature absence problem*. Type I is apparently fatal for any meaningful prediction, as there is not *any* feature at all for the model to utilize. Type II is also disastrous for a sensible prediction, as the unigram feature C0 alone could hardly determine the label or the role of the character correctly. In Chinese, the majority of character may occur in any position of a word,

i.e. its label can be either **S**tart, **M**iddle, **E**nd or **S**ingleton, except for a few characters which have dominant roles such as prefix (e.g. 非, 'not/non', 反 'anti') or suffix (e.g. 者 'one who does or is ...', 化 a verbalizing suffix). Note that even for these characters, there are ambiguities as for the role in a word, e.g. 非 can be the end of a word as in 是非 'right and wrong/quarrel'.

Character instances in Type III have a better chance of being correctly labeled by the model but relying only on the bigram context on one side is likely to be of high bias in the first place, and it might be the case that the bigram context on "the other side" is more informative than the one that are seen in the training corpus. Moreover, the association of a certain character co-occurrence with a certain label in the training corpus might also be merely by chance, especially for those co-occurrences that are less frequent in the training corpus. Finally, the statistics here is with regard to characters, and we need be aware that the recognition of an OOV fails even if only one the character is incorrectly labeled, which means this 67% feature absence case may explain a much higher percentage of OOV tokens that are not recognized.

It is clear that the issue above is an inherent problem of the feature space and is out of reach for the clever optimizations offered by machine learning algorithms. To illustrate this, we fit the discussion in an abstract view of classification algorithms in machine learning. The model can be viewed as hyper planes that separate the feature space, in which the training instances are dots. The separation should be made in such as a way that instances of the same class are in the same subspace, if noise are not taken into account. The prediction or testing process is fairly straightforward once these hyper planes are determined in the training process. For an new/unseen instance, its features corresponds to coordinates of dimensions in the space, once the coordinates are determined, the instance fits an area, preferably a dot, in the space separated by the model. The subspace that the instance falls in defines its label. However, the situation in the feature absence problem is that very few, or in extreme cases, no coordinates are given for the new instance in testing data, thus the area in the space determined by these coordinates are so vast that they may cross the boundaries of the hyper planes. In this case, one would not be able to tell which subspace or class that instance belongs to. Of course, sequence labeling is more complicated than classification, but the above argument also holds.

1.4 Zipf's law and its implications

The problem seems to be that our training corpus is too small to contain all the bigram co-occurrences that occur in the testing corpus. So can we simply enlarge our training corpus to solve this problem? Unfortunately, there are two factors that make this proposal less appealing as at the first glance. Firstly, the training corpus is obtained via human annotations, which are expensive. Secondly, empirical study shows that the scale of corpus that we need to capture enough features grows at an exponential rate with regard to the number of distinct features (Zhao et al., 2010). The second factor is determined by the Zipf's law (Zipf, 1949), which widely applicable to linguistics data and empirical distributions in many other areas.

Zipf's law states that the frequency of an item (character, word, bigram, etc.) is approximately equal to the inverse of its rank in frequency, which can be expressed by the following formula:

$$f = C/r, \tag{1}$$

where C is some constant, f is the frequency of the item and r is its rank of frequency in the set of the item. A perfect Zipfian distribution would be a straight line of slope -1, with the axes being log of word frequency and the log of word rank. The empirical usually have minor deviation from the perfect scenario (Figure 1). There are many vocabulary studies that report Zipf's law in various language and genres (Baroni, 2008). The distribution of Chinese characters ngrams and word ngrams are of no exception.

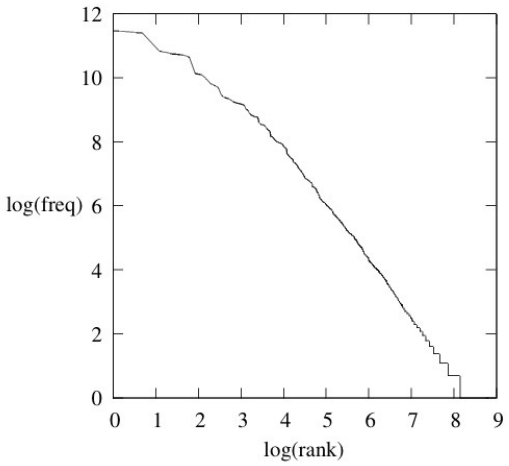


Figure 1: A Zifp's law curve of word frequency

One immediate implication is that only a small percentage of items occur very frequently while the majority of items occur very rarely (in extreme but common cases, the frequency equals to 1) in real texts. On the word level, Zipf's law suggests that given a relatively small sample of the sentences in a language such as a treebank corpus, most words are so rare that they are likely to occur in either training section or testing section but *not* both, which explains why OOVs occur in the first place.

On the character level, Zipf's law means that most character co-occurrences are so rare that they occur either in training or testing corpus but not both and only a few character co-occurrences are so frequent that they are likely to occur in both corpus. This explains why very heavy feature pruning works, as a small subset of all character co-occurrence types take account for the most co-occurrence tokens. In short, the feature absence problem is a rule rather than an exception.

Another interpretation of Zipf's law is that it predicts how large a corpus is needed to cover a certain number of distinct word/character ngrams. The general idea is that

since a few items occur very frequently, a new non-frequent item will only appear after seeing many occurrences of these frequent items. Mathematically, the sum of all relative frequencies in a Zipf distribution is equal to the harmonic series and therefore:

$$\sum_{n=1}^{\infty} \frac{1}{n} = \infty. \quad (2)$$

This formula states that series gets arbitrarily large as n becomes larger, which suggests that exponential more tokens have to occur before more distinct types are encountered. This has been confirmed by empirical study (Zhao et al., 2010) as well. The bad news is that even though the scale of commonly seen Chinese characters is only at thousands. The word formation process that combines characters is very dynamic and productive. Even if we only consider words that are made of two characters, the upper bound of number of extinct types is 10^6 ($10^3 \times 10^3$). Although the actual number of distinct two-character words is far smaller than the upper bound, the scale of annotated corpus needed to solve the data sparseness problem is still tremendous. Given the inevitable presence of the feature absence problem, which is governed by Zipf's law, the efforts on solving the OOV recognition problem by applying stronger machine learning algorithms or smarter system combination are beneficial yet seem to aim only at the tip of the iceberg.

Relevance to Language Acquisition

Before we move to the discussion of possible solutions of the OOV recognition problem, let us first examine an interesting connection between the limitation of feature-based machine learning approaches to CWS and the drawbacks of the item-based approach to language acquisition.

Since Chomsky (1965), linguists have been aware of the distinction between competence and performance, which suggests that it is limited to draw conclusions only from observed linguistic data. For example, some words have never been said but are nevertheless grammatically correct. This distinction has also been widely accepted in the subfield of language acquisition, even by researchers that do not follow the generative grammar. However, this idea has been recently challenged by the *item or usage based theory* of language acquisition (Tomasello, 2000; Hay & Baayen 2005, etc). The item-based approach states that language acquisition can be achieved by memorizing and operating specific schemas of linguistic forms and constructions, in contrast with the traditional thought of learning grammar rules that consist a productive/generative linguistic system.

Note that the claim of the item-based approach to language acquisition is similar to the feature-based machine learning approaches to CWS at an abstract level. Both approaches build models using specific surface linguistic forms and their co-occurrences and the model retrieves such stored "pairings of form and functions" to do the production or recognition, although the models in the former do not necessarily of statistical nature as those in the latter do.

Interestingly, the generative school fights back (Young 2011) and argues that there are some inherent limitations in the item-based approach, as Zipfian distribution determines that most "pairings of form and functions" will never be heard and even for those do occur may be so infrequent that the storage of usage of such pairings is not reliable. Further empirical study has shown that the item- based approach is not supported by statistical evidence in language acquisition data. On the contrary, generative grammars are consistent with empirical data, based on a model that considers the interaction of Zipfian distribution and the combinations of linguistic items.

While CWS is a different domain than language acquisition, the arguments here may still provide a hint on understanding the OOV problem. It is likely that the Zipfian nature of character/word ngram distributions ensures that the overlap of these surface form co-occurrence based features in training and testing corpus of CWS systems are quite low by type unless the corpus size is very large, which unfortunately requires an exponential growth of the size of the annotated corpus. And the consistency of empirical data with generative grammars that have been observed in language acquisition case studies may also hold in the word formation process of Chinese, which implies an alternative formalism for solving CWS problem in general and OOV problem in particular.

Generative Word Formation Model

The idea that word formation in Chinese is an generative system is reasonable in both language acquisition and theoretical linguistics. This Morphology of Chinese, which is represented by early works such as (Zhao, 1968; Lü, 1979) and more recent work in the framework of generative linguistics such as (Huang, 1984; Dai, 1992; Duanmu, 1997; Packard, 2000; Xue, 2001).

Dai (1992) introduced the idea that different notations of wordhood co-exist, including morphological word, syntactic word and phonological word. The interactions between them explain various word formation phenomena. But his model is basically a static lexicon, which does not provide a concrete proposal on how morphological words are derived.

Packard (2000) is probably the most influential modern work, which treats the morphology as an extension of syntax below the word (X_0) level, following the thinking of Selkirk (1982). Packard (2000) is based on the "form class description", which assigns words and their components (characters) part-of-speech like tags called form class. He has also suggested so called "Headness Principle", which states that nouns have nominal components (characters) on the right and verbs have verbal components (characters) on the left. Like Dai (1992), Packard (2000) also fits into a lexicalism framework, and considers both morphemes and complex words with their "precompiled" morphological structures in the lexicon, except for complex words containing grammatical affixes.

In contrast, Xue (2001) have proposed a system that derives virtually *all* the complex words *using syntax rules* or in the morphology module after syntactic analysis, following the theory of distributed morphology (Halle & Marantz 1993,

1994). The boundary of syntax and morphology further blurred and the operation scope of syntax rules expand to most parts of the morphology.

Despite the disagreements, both Packard (2000) and Xue (2001) agree that part-of-speech like tags for characters and words and syntactic or morphological rules that describe the derivation of these tags make essential parts of a *generative word formation system* for Chinese. Computational linguists have started rethinking the limitations of feature based machine learning approach for CWS and has called for morphology-based analysis of OOVs (Dong et al., 2010). Furthermore, there are already pilot works in this direction, such as Zhao (2009), Li (2011) and Ma et al. (2012). Both methods happen to be formulated as learning a joint model for segmentation and parsing, which has certain practical advantages, but is not necessary for learning a word formation model.

Zhao (2009) has proposed a character-based dependency parsing model, in which the word formation is formulated as the in-word character dependencies, without any part-of-speech tags or dependency labels. The dependency model has comparable performance on the CWS task as the state-of-the art sequence labeling based segmenters. While it is an interesting investigation, pure character-wise dependencies seem to be inadequate to model the word formation process in a general and productive manner.

Li (2011) has proposed a unified parsing model that can parse both word structures and phrase structures. Part-of-speech tags and constituent labels are utilized in this model. The model extends probabilistic context free grammar based constituent parsing to handle the inner structure of words, which has a flavor of generative word formation model, i.e. syntactic rules are used to analyze the word formation process. The performance of this model on CWS task is slightly better than the state-of-the-art but no significant improvement on OOV recognition has been reported. Note that this work makes a distinction between flat words and non-flat words and the grammar model only deals with the generation of the non-flat words. Here the non-flat words are defined as those words that contain productive suffix and/or prefix, which is only a small subset of words that can be possibly analyzed by syntactic or morphological rules. In this sense, Li (2011) can be viewed as an implementation of Packard (2000). The model's low coverage of the word formation phenomena may explain why this model has not brought advancement on OOV recognition. The morphological model might be more powerful on OOV recognition, if syntax-like rules were used to analyze most of, rather than a small portion of, complex words, i.e. by implementing Xue (2001). Nevertheless, the results presented in Li (2011) are encouraging, as it has shown the effectiveness of analyzing word formation using generative rules. Note that Li (2011) follows a standard paradigm in modern syntactic parsing: the probabilistic syntax model that is used for parsing is learned from an annotated treebank. So far, we have also limited our discussion to this default.

Ma et al. (2012) have proposed a semi-automatic approach to Chinese word structure annotation. They have argued that Li (2011) only annotated affixations, which only covered 35% of word types in the corpus and was insufficient to deal with the OOV problem. In contrast, their annotation has covered more morphological

phenomena, including compounding, which is a more popular word formation process in Chinese. Unfortunate, the usefulness of such annotation for the OOV problem has not been validated by experiments yet.

One may wonder whether it is possible to have such a strong machine learning algorithm that can overcome the limitations of current learning algorithms used in CWS and effectively induce the word structure without the explicit notion of word formation model and the utilization of manual treebank annotation. This turns out to be quite a difficult task, and the current computational learning research under the framework of Probably Approximately Correct (PAC, Valiant, 1984) suggests that it is virtually impossible to learn languages such as finite state and context free language, given only distribution of surface forms (Yang, 2011). But learnability results are in a general sense and can be modified, e.g. adding certain assumptions, to suit various learning scenario, which is an interesting topic itself.

Conclusion

In this paper, we have reviewed some state-of-art methods for Chinese word segmentation, with a focus on the role of distributional evidence and feature-based machine learning algorithms. By showing the Zipfian nature of the distributional evidence, we have further investigated the limitations of feature-based statistical machine learning models for CWS, which can be summarized as the feature absence problem. Drawing the connection with language acquisition literature, we have speculated that a generative linguistic system may help overcome the limitations of current methods. This speculation is supported by some formal linguistic analysis of Chinese morphology. Finally, we have shown that recent results in relevant computational modeling suggests that it is indeed a promising direction to investigate generative word formation models in order to come up with better CWS system.

References

1. Marco Baroni. 2008. Distributions in text. In Lüdelign, A. & Kytö, M. (Eds.) *Corpus linguistics: An international hanbook*. Mouton de Gruyter, Berlin, Germany.
2. Noam Chomsky. 1965. Aspects of the theory of syntax. MIT Press, Cambridge, USA.
3. Xiang-Ling Dai. 1992. *Chinese Morphology and its Interface with the Syntax*. PhD Dissertation, Ohio State University.
4. Zhendong Dong, Qiang Dong and Changling Hao. 2010. Word segmentation needs change - from a linguist's view. In Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing. Beijing, China.
5. San Duanmu. 1997. "Wordhood in Chinese", in Jerome J. Packard ed. *New Approaches to Chinese Word Formation*. Mouton de Gruyter, New York, USA.
6. Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of Forth SIGHAN Workshop on Chinese Language Processing*. Jeju Island, Korea.
7. Yoav Goldberg and Michael Elhadad. 2009. On the role of lexical features in sequence labeling. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore.

8. Morris Halle and Alec Marantz. 1993. Distributed Morphology and the pieces of inflection, in Hale, Kenneth and Samuel Jay Keyser eds. *The View from Building 20*. The MIT Press, Cambridge, USA.
9. Morris Halle and Alec Marantz. 1994. Some key features of Distributed Morphology. *MIT Working Papers in Linguistics* 21, 275-288.
10. Jennifer Hay and Harald Baayen. 2005. Shifting paradigms: gradient structure in morphology. Trends in Cognitive Sciences, 9, 342-348
11. James C. T. Huang. 1984. Phrase structure, lexical integrity, and Chinese compounds. *Journal of the Chinese Language Teachers Association* 19.2:53-78.
12. Wenbin Jiang, Liang Huang, Qun Liu, Yajuan Lu. 2008. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL 2008: HLT*. Columbus, USA.
13. John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*. Williamstown, MA, USA
14. Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. Sydney, Australia
15. Zhongguo Li. 2011. Parsing the internal structure of words: a new paradigm for Chinese word segmentation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA.
16. Nanyuan Liang. 1986. On computer automatic word segmentation of written Chinese. *Journal of Chinese Information Processing*, 1(1).
17. Shuxiang Lü. 1979. *Hanyu Yufa Fenxi Wenti "Problems in the analysis of Chinese grammar"*. Shangwu Yinshuguan, Beijing, China.
18. Jianqiang Ma, Chunyu Kit and Dale Gerdemann. 2012. Semi-automatic annotation of Chinese word structure. In the Proceedings of *2nd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, Tianjin, China.
19. Jerome Packard. 2000. The Morphology of Chinese: A Linguistic and Cognitive Approach. Cambridge University Press, Cambridge, UK.
20. Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING*. Geneva, Switzerland.
21. Elisabeth O. Selkirk. 1982. The Syntax of Words. Cambridge, Massachusetts: The MIT Press, Cambridge, USA.
22. Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word- segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377-404.
23. Weiwei Sun. 2010. Word-based and character-based word segmentation models: Comparison and combination. In *Proceedings of the 23rd International Conference on Computational Linguistics*: *Posters Session*. Beijing, China.
24. Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, USA.
25. Weiwei Sun and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, UK.

26. Michael Tomasello. 2000. First steps toward a usage-based theory of language acquisition. *Cognitive Linguistics*, 11, 61-82.
27. Leslie G. Valiant. 1984. A theory of the learnable. *Communications of the ACM*, 27, 1134-1142.
28. Kun Wang, Chengqing Zong and Keh-Yih Su. 2010. A character-based joint model for Chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China.
29. Jonathan Webster and Chunyu Kit. 1992. Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on*. Nates, France.
30. Nianwen Xue. 2001. Defining and automatically identifying words in Chinese. Phd Thesis, University of Delaware.
31. Nianwen Xue. 2003. Chinese Word Segmentation as Characater Tagging. *Computational Linguistics and Chinese Language Processing*, 8(1): 29-48
32. Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2)207-238.
33. Charles Yang. 2011. A computational models of syntactic acquisition. *Wiley Interdisciplinary Reviews: Cognitive Science.*
34. Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08: HLT*. Columbus, USA.
35. Hai Zhao. 2009. Character-level dependencies in Chinese: usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the ACL.* Athens, Greece.
36. Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An Improved Chinese Word Segmentation System with Conditional Random Field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. Sydney, Australia.
37. Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of word segmentation and named entity recognition. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing* (SIGHAN-6). Hyderabad, India.
38. Hai Zhao and Chunyu Kit. 2009. A simple and efficient model pruning method for conditional random fields. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*. Springer-Verlag Berlin, Heidelberg, Germany.
39. Hai Zhao, Yan Song and Chunyu Kit. 2010. How Large a Corpus do We Need:Statistical Method vs. Rule-based Method. In *Proceedings of LREC-2010*. Malta.
40. Hongmei Zhao and Qun Liu. 2010. The CIPS-SIGHAN CLP 2010 Chinese Word Segmentation Bakeoff. In *Proceedings of the First CPS-SIGHAN Joint Conference on Chinese Language Processing*. Beijing, China.
41. Yuen-Ren Zhao. 1968. *Grammar of Spoken Chinese*. University of California Press, Berkeley and Los Angeles, USA.
42. George Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* Addison-Wisley. Oxford, UK

Named Entity Filtering Based on Concept Association Graphs

Oskar Gross[1], Antoine Doucet[2], and Hannu Toivonen[1]

[1] Department of Computer Science
P. O. Box 68 (Gustaf Hällströmin katu 2b)
FI–00014 University of Helsinki
Finland
first.last@cs.helsinki.fi
[2] Normandy University – UNICAEN
GREYC, CNRS UMR–6072
F–14032 Caen Cedex
France
first.last@unicaen.fr

Abstract. In this paper, we introduce a novel technique for named entity filtering, focused on the analysis of word association networks. We present an approach for modelling concepts which are distinctively related to specific named entity. We evaluated our approach in the context of the TREC Knowledge Base Acceleration track, and we obtained significantly better performance than the top-ranked systems. For this task, given the set of all named entities and nouns, our approach proved better-performing for named entity filtering than the baseline SVM classifier. This performance is the result of the ability to disambiguate entities, by taking into account the concepts relevant to a specific named entity.

1 Introduction

In this paper we will demonstrate a method for detecting documents which are related to a target named entity. The complexity of the task lies in the fact, that a named entity and a document could be related even when the named entity is not explicitly mentioned in the document. Moreover, it is possible that two absolutely different entities have the same name (e.g. *Queen* might refer to a British rock band, a British women's magazine or a subway station in Toronto).

Knowledge bases (e.g. Wikipedia) collect, structure and validate information about certain entities or events. At the moment articles in the knowledge bases are managed by humans and new information is added to the article with some delay. According to Frank et al. [1] the median of the updates delay in Wikipedia is 356 days. Detecting automatically news stories, which are novel and relevant to Wikipedia articles would decrease human labour a lot. In this paper we are focusing on detecting documents, which are relevant to a news story and omit the novelty aspect. In addition to knowledge base acceleration, some examples of other potential applications are media monitoring, topic mining and advertising.

We propose a graph based method for relating documents to target named entities. The fundamental idea of the method is to model a named entity by analysing its co-occurring concepts. We will provide a methodology for creating named entity specific graphs, which we use for filtering documents. We will evaluate our methodology by using data provided by NIST during the TREC *Knowledge Base Acceleration* (KBA) track in 2012. The main motivation for this task is to detect documents about entities, which may contain information to be added to the knowledge bases.

The rest of the paper is organized as follows: in the next section we will introduce the related work. In Section 3 we will introduce the method for generating concept graphs. How to filter the documents using the proposed method will be described in Section 4. We evaluate our method and compare its results to the state of art in Section 5. Finally, conclusions are drawn in Section 6.

2 Related Work on Named Entity Filtering

Named-entity filtering, from a stream of news data, is related to several fields where discovering and following-up on events concerning a given topic is especially valuable. In all these fields, the ability to identify named-entities is an essential performance enhancer.

Followingly, this task concerns diverse fields of information retrieval, such as news surveillance [2], entity linking [3] and text categorization [4]. In this section, we will focus on the closest and most significant papers, notably on the approaches developed during the recent TREC KBA track, whose first round in 2012 [1] focused specifically on the task of named-entity filtering.

News Surveillance. The task of news surveillance is to give alerts for all the events related to a given domain of interest. For instance, health agencies (e.g., the World Health Organization) wish to be informed of every case of occurrence of a transmittable disease, as close as possible from the moment when it occurred [2]. Other typical fields of application lie in the field of intelligence, and in finance, where the era of high frequency trading turned the apprehension of news milliseconds earlier into a decisive advantage. However, most approaches are strongly domain-dependent, requiring thousands of syntactic patterns to detect relevant news alerts [5].

Entity Linking. Entity Linking is the task of automatically linking phrases occurring in a document to entries in a knowledge base. Several comparative evaluation competitions have run in the recent past, testifying on the great progress achieved (INEX's Link-the-Wiki [6], Text Analysis Conference's Knowledge Base Population (KBP) [7]). Entity linking is nowadays a well-understood problem, that paves one way leading towards named-entity filtering : once the named-entities are marked within a text, it "only" remains to compute the centrality and relevance of the named entity: is it the main topic of the document, or is it simply mentioned?

Many of the methods presented in the TREC KBA track follow up from entity linking. This is natural, since the corpus was provided with pre-extracted named-entities.

Liu and Fang [8] presented one of the best performing approaches of the KBA track, by building "entity profiles". By fetching a snapshot of the Wikipedia, and considering the anchor text of all internal Wikipedia links as related entities, they defined a wider representation of named entities.

Araujo et al. [9] underlined that 4% of the Wikipedia citations do not mention the Wikipedia they are cited by. This motivates their focus on the detection of documents that do not mention a named entity that is yet central to it. To achieve this, they fed their model with the Google Cross-Lingual Dictionary (GCLD) [10], a ready-made resource associating Wikipedia entries to strings. As the TREC KBA topics are named-entities for which a Wikipedia entry is defined, they could replace the topics with the strings returned by the GCLD. With adequate parameters, the technique obtained the best performance for centrality and relevance.

Text Categorization. Text categorization is the task of assigning categories to a text, given a training set of text-category assignments. Text filtering is the special case when there is only one category, and the classifier is only to decide whether a given text belongs to it, or not. Such a categorization is usually led based on word term features, and the best-performing technique in the state of the art is the well-known SVM [11].

Kjersten and McNamee [12] hence proposed to filter the document sets, using the SVM classifier over a set of features composed of the named entities provided by the TREC KBA organizers. Positive examples from the training set were those marked as central. All the others were considered negative. The technique proved that this was achievable, and it obtained the best and second-best performance (out of 40 runs) for centrality.

Other approaches. The approaches presented at TREC KBA 2012 can essentially be split into two categories [1]: those that exploit rich features from a Knowledge Base (Wikipedia or Google Dictionary) and those that focus on machine learning techniques (such as SVM).

Unlike the approaches from the first category, our technique is endogenous, that is, it does not make use of any resources that are not present in the corpus. Hence, it can easily generalize accross domains and languages (even though, the latter was not yet verified).

To the best of our knowledge, no recent techniques have been proposed that would rely on the construction and exploitation of concept association graphs. The closest example was introduced by Gamon [13]. He adressed the problem of novelty detection by building an association graph connecting sentences and sentence fragments, and chose to exploit a number of graph-based features that were assumed to be good indicators of novelty. The method tied with the best techniques presented in the TREC novelty track 2 years earlier [14], but the authors himself questioned the significance of the improvement.

In the following sections, and in the light of related work, we will introduce our approach in full details.

3 Named Entity Modelling

Our method is based on the idea, that a news item is related to a named entity when both of them are related to the same concepts. Thus our approach consists of two steps:

1. We calculate which concepts are related to each other, building an association graph of named entities;
2. For each news story, and for each named entity in a query (or TREC topic), we calculate the overlap between the concepts related to the named entity and those related to the news story.

In the rest of this section, we will detail, in chronological order, the different steps in which we process document stream data so as to build our background concept association graph.

3.1 Selecting Concepts

To build our concept graph, the very first step is naturally to select the concepts. We will use TREC data which contains part-of-speech tags, lemmatized forms of the words and is annotated with Stanford NER [15]. The annotation of Stanford NER identifies whether each single word is itself or is part of a named entity and tags it with a type. For the data to fit our purposes, we post-processed the resulting set of named entities as follows :

1. Concatenate each named entity parts with an underscore (adjacent words with the same type (organization, person etc) are concatenated together);
2. Remove all words which are not nouns.

We extract nouns and named entities from the documents and discard everything else. This choice is motivated by nouns and named entities being conceptually more basic than concepts referred to by verbs or prepositions [16]. Of course, we lose some information with this step. As a final step, we lower-case and leave the lemmatized form of the words.

3.2 Building the Concept Graph

We build the concept graph, based on learning data, containing annotations indicating which named entities and which news stories are truly related. The graph generation consists of two steps: (1) calculating the co-occurrence graph using the documents and (2) cleaning the graph, by removing unnecessary edges and nodes. We will next give an overview of the graph construction, which is follows the same as principles as in Gross et.al in [17].

The first step is based on log-likelihood ratio calculation. Consider the set of documents, which are connected (by annotation) to named entity n, by $d \in C_n$. We will consider a document d as bag of sentences S_d and each sentence as bag of words $T_d \in S_d$. The set of all words is $T = \bigcup T_d$.

We analyze word co-occurrences on the granularity of sentences, since words which are in one sentence generally have a stronger relation to each other [18].

The concept graph $G_n = (V_n, E_n, W_n)$ is a weighted, undirected graph with nodes V_n, edges $E_n \subset V_n \times V_n$, and edge weights $W_n : V_n \times V_n \to \mathbb{R}_+$. For notational convenience, we assume $W(e_1, e_2) = 0$ if there is no edge between e_1 and e_2.

Construction of the graph then starts by using all terms in the corpus C_n as nodes, i.e., $V = T$.

We use the log likelihood ratio (LLR) to measure the strength of an association between two terms [19]. In [17] we showed that the co-occurrences, as measured by LLR do make sense, though other word association measures would probably be equally suitable.

LLR measures how much the observed joint distribution of terms x and y differs from their distribution under the null hypothesis of independence, i.e., how strong is the association between them. Edges are constructed for term pairs $\{e_1, e_2\}$ in T that have a strong log-likelihood ratio $LLR(e_1, e_2)$.

In other words, we in principle compute LLR for the union P of all the pairs of terms in all sentences of the corpus:

$$P = \bigcup_{d \in C_n} \bigcup_{s_d \in d} s_d \times s_d. \tag{1}$$

3.3 Cleaning up the Graph

The goal of the graph cleaning process is to remove edges and nodes which, at this point, we find unnecessary. We are interested in leaving only such associations, which are directly related to named entities. For this, we define N as the combinations of the different parts of each named entity. Consider a named entity "Annie_Laurie_Gaylor". For this named entity the possible combinations are N = {"Annie_Laurie_Gaylor", "Annie_Laurie", "Annie_Gaylor", "Laurie_Gaylor", "Annie", "Laurie", "Gaylor"}. In the next step we leave only nodes which are associated to the parts of the named entity, i.e. $e_1 \in N \vee e_2 \in N$.

Our experience showed, that there are nouns, which appear in all the named entity graphs. For reducing some amount of the noise, we remove all such nodes, which appear in all the different named entity graphs. Let Γ denote the set of all named entity graphs. Then we will construct a set of nodes which are found in all graphs as

$$U = \bigcap_{G_n \in \Gamma} V_n(G_n).$$

These nodes will be removed from all the graphs:

$$G_n = (V \setminus U, \{e \in E : e_1 \notin U \wedge e_2 \notin U\}, W).$$

In the next section we will show, how we utilise these graphs for detecting the news stories which are related to a given topic.

4 Document Filtering with Concept Graphs

4.1 Principles

To be able to rank documents with respect to the NEs of any given TREC KBA topic, consisting of one or more named entities, it remains to design a way to compute relevance scores based on our graph model.

We do so by relying on the concept of word co-occurrence, with the following principles in mind, on what we expect a more interesting document to be like. First it is reasonable to assume, that the concepts in the document should intersect with the concepts which are strongly related to the named entity. On the other hand, as the named entity could appear in many different contexts (e.g. the president of the USA could be related to financial, political, arts & entertainment topics et cetera), we should not penalize a document for *not* being related to some neighbours.

4.2 Document Relevance Evaluation

To calculate the relevance score of a given document, w.r.t. a given TREC KBA topic, we proceed as follows. We post-process the document exactly as we described in Section 3.1 and calculate the named entity specific graph models. Then we use the weights in the named entity graphs to calculate the relevance of a document.

As the first part is covered in the previous section, we will hereby describe the second step.

Documents relevance is calculated by measuring how strongly words in the document are connected to the named entity of interest. Let us consider the target document d_t which contains the words w_t.

For a named entity n we calculate the relevance status value RSV for an incoming document, given the entity graph G_n as:

$$RSV(G_n, d_t) = \frac{1}{|w_t|} \sum_{w \in w_t} \sum_{v \in V_n} W(w, v)$$

which is the average edge weight of the words in the named entity graph. The rationale for calculating RSV as the average of the edge weights is to reduce the impact of outliers. Indeed, we believe that averaging over all the edges represents better the *general* match between the document and the named entity, than, e.g., summing up or taking the maximum weight. This is the case when the overlap between the document and the named entity graph is small (e.g. 3 or 4 nodes) and one node is strongly connected to the entity graph and other nodes are weakly connected to the named entity.

5 Evaluation

In this section we will describe the evaluation methodology and present the subsequent results.

5.1 Method

The KBA 2012 evaluation data consists of 57,750 human-generated judgements rating the relevance of documents to target entities. The KBA stream corpus of 462M documents covers 4,973 contiguous hours of news, blogs, and forum posts. It includes dozens of languages beside English. In our task we use a subset of the data – documents with a reasonable chance to be written in English that have been automatically POS tagged. This subset contains roughly 367M documents.

The data spans over 8 months - from October 2011 till May 2012. The data is divided into two sets by using a *cut-off* date, which is January the 1st. Documents published before the cut-off date are used as the training set and documents after the cut-off date form the test set.

Each article in the annotation set is scored in two categories - *relevant* and *central*. For each entity, the document *central* score is high if the respective entity is the central topic of the document. The *relevant* score is high if the document is indirectly relevant to a certain named entity. In the evaluation no pooling of the TREC participants results is used.

The accuracy of the method is calculated on the test set by using the articles which are annotated. However, in the scoring phase, the algorithm does not know which articles are annotated and not, and it therefore needs to go through all 367M documents.

The methods are evaluated by using the standard information retrieval measures – precision, recall and their harmonic mean F_1.

5.2 Graph Based Model

The graph models for each named entity are created by using the methodology described in Section 3. The similarity score for each document and named entity pair is calculated by using the method given in Section 4.

Baseline

For comparing our methodology, we will use a standard machine learning algorithm, support vector machines (SVM) as a baseline. SVM has been shown to be successful in text categorization [20] and document filtering is a special case of text categorization with two categories.

For each named entity we calculate the entity specific SVM model by using the annotation data. We use the same feature-set as we use in the graph based models. In total there are 13,111 features. The features are used as binary features, representing whether a word is found in the document or not.

We carried out our SVM experiments with SVM light toolkit [11]. Following good data analysis practices, we divided the training set into two parts – 80% of the documents as a training set and 20% of the news into test set. We used these two sets to estimate the performance of our method and also to analyze SVM scores for different documents. We observed that the meaningful values for SVM classification were between -2 and 2, thus we scaled this range to be between 0 and 1000.

The SVM models used for scoring after the cut-off date were trained on the whole dataset which was published before the *cut-off* date (i.e. 1st of January 2012).

Results

For measuring the performance of our method, we will compare it to the best scoring methods at the TREC 2012 KBA track – `CWI-google_dic_31` proposed by Araujo et al. [9] and `hltcoe-wordNER` proposed by Kjersten and McNamee [12]. `CWI-google_dic_31` uses Google Cross Lingual dictionary and the `hltcoe-wordNER` uses SVM for document filtering, very similarly to our baseline.

For measuring, whether the accuracy of the different methods is significant, we approximated the pairwise p-values between performance values of the methods. The statistical significance is calculated by using two samples – for each method we chose the cutoff value for which the average F_1 measure was maximized. The sample consists of all the topics F_1 measures for respective cutoff. We then use the Wilcoxon Rank Sum test for testing whether the difference between samples is significant.

The results are summarized in Table 1. We can observe that we are doing better than the baseline methodology in both categories. In the *central* category our method is a bit worse than the best performing method of the KBA track in TREC (`hltcoe-wordNER`), although not significantly. On the other hand, our methodology is the best perfomer in the *central+relevant* category by beating `CWI-google_dic_31`. The reason for such good accuracy in this category can be related to the approach of our methodology – we put strong emphasis on the context, where the named entity appears.

While our graph-based approach outperforms all the other methods in the *central+relevant* category, it is important to emphasize that a strength of our method is that it is resource free and relies only on the corpus. It should hence be much easier to generalize to other data sets, e.g., written in other languages.

Future Work

One of the current shortcomings of our graph based models is that they are static. Intuitively it is reasonable to assume that the accuracy of the graph models decreases over time, as the topics about a certain entity may also change over time. As an example one could consider a sportsman, who at different times, can be related to documents about Olympics, gold medals and doping.

Table 1. The F_1 measures for central and central+relevant. The asterisk marks the F_1-measures which are significantly ($p < 0.05$) different (in the same category) from the graph based method.

Method	**Central**	**Central+Relevant**
Baseline (SVM)	0.327	0.569*
CWI-google_dic_31	0.291	0.637
hltcoe-wordNER	**0.359**	0.494*
Named Entity Graph	0.341	**0.691**

To test the hypothesis, we calculated the precision of our method over time by using the test set annotation files. The result can be observed in Figure 5.2.

As shown in the figure, the precision is high in the beginning, but it gets lower over time, suggesting that our model becomes more and more outdated. Observe, that the drop from 0.4 to ≈ 0.3 is rapid, approximately two weeks, and after this the precision decreases in a more constant rate. This observation is a good hint for pointing our future work towards updating the graph models while analysing the stream.

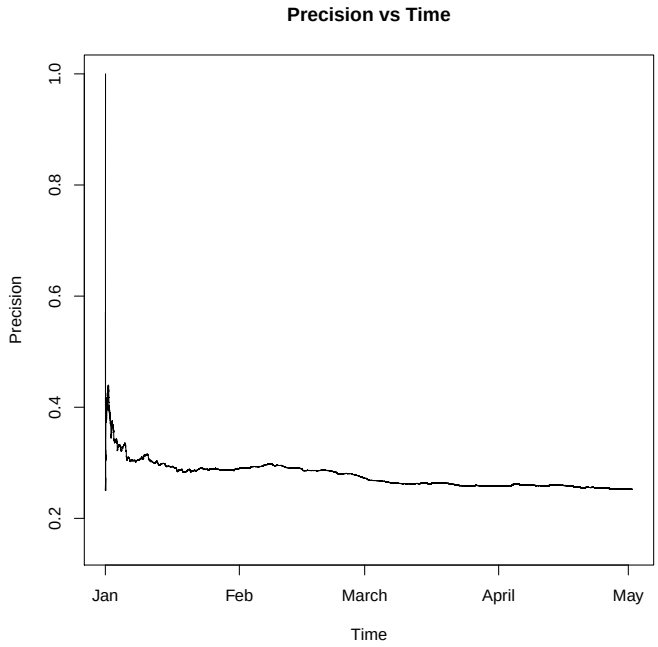


Fig. 1. Decrease of the precision of the graph based method over time.

6 Conclusion

In this paper we have demonstrated a corpus based approach for modelling named entities. When designing the method, we have considered the following aspects. First, we find it important to be independent from language as much as possible and we use fairly simple methods in order to get rid of some of the obvious noise. Secondly, we have chosen a graph based approach due to the interpretable and easily expandable nature of the models.

We have implemented and experimented with our approach with encouraging results. Immediate future work will focus of implementing the update of the background graph, which is currently static, based only on the training data. In context of stream data, using static models is inadequate. The main aspect of the named entities is, that they evolve in time. We believe that taking into account the temporal aspect of the named entities could give a lot of improvement of the performance of the system.

Acknowledgements: This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland.

References

1. Frank, J.R., Kleiman-Weiner, M., Roberts, D.A., Niu, F., Zhang, C., R, C., Soboroff, I.: Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of the 21st Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, November 6-9, 2012, National Institute of Standards and Technology (NIST) (2012)
2. Linge, J., Steinberger, R., Weber, T., Yangarber, R., van der Goot, E., Al Khudhairy, D., Stilianakis, N.: Internet surveillance systems for early alerting of threats. Eurosurveillance **14** (2009)
3. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, y.H., Falco, A.O., eds.: CIKM, ACM (2007) 233–242
4. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34** (2002) 1–47
5. Steinberger, R.: A survey of methods to ease the development of highly multilingual text mining applications. Language Resources and Evaluation (2011) 1–22
6. Huang, D.W., Xu, Y., Trotman, A., Geva, S.: Overview of INEX 2007 link the wiki track. In Fuhr, N., Kamps, J., Lalmas, M., Trotman, A., eds.: Focused Access to XML Documents. Springer-Verlag, Berlin, Heidelberg (2008) 373–387
7. Ji, H., Grishman, R.: Knowledge base population: successful approaches and challenges. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 1148–1158
8. Liu, X., Fang, H.: Entity Profile based Approach in Automatic Knowledge Finding. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of the 21st Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, November 6-9, 2012, National Institute of Standards and Technology (NIST) (2012)

9. Araujo, S., Gebremeskel, G., He, J., Bosscarino, C., de Vries, A.: CWI at TREC 2012, KBA Track and Session Track. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of the 21st Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, November 6-9, 2012, National Institute of Standards and Technology (NIST) (2012)
10. Spitkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for english wikipedia concepts. In Chair), N.C.C., Choukri, K., Declerck, T., Doan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., eds.: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, European Language Resources Association (ELRA) (2012)
11. Joachims, T.: Making large-scale support vector machine learning practical. In Schölkopf, B., Burges, C.J.C., Smola, A.J., eds.: Advances in kernel methods. MIT Press, Cambridge, MA, USA (1999) 169–184
12. Kjersten, B., McNamee, P.: The HLTCOE Approach to the TREC 2012 KBA Track. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of the 21st Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, November 6-9, 2012, National Institute of Standards and Technology (NIST) (2012)
13. Gamon, M.: Graph-based text representation for novelty detection. In: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing, New York City, Association for Computational Linguistics (2006) 17–24
14. Soboroff, I.: Overview of the trec 2004 novelty track. In Voorhees, E.M., Buckland, L.P., eds.: TREC, National Institute of Standards and Technology (NIST) (2004)
15. 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. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 363–370
16. Gentner, D.: Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. BBN report; no. 4854 (1982)
17. Gross, O., Toivonen, H., Toivanen, J.M., Valitutti, A.: Lexical creativity from word associations. In: Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on. (2012) 35 –42
18. Miller, G.: Wordnet: a lexical database for english. Communications of the ACM **38** (1995) 39–41
19. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational linguistics **19** (1993) 61–74
20. Joachims, T.: Text categorization with suport vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning. ECML '98, London, UK, UK, Springer-Verlag (1998) 137–142

A New Approach to Tagging in Indian Languages

Kavi Narayana Murthy and Srinivasu Badugu

School of Computer and Information Sciences,
University of Hyderabad, India
knmuh@yahoo.com,srinivasucse@gmail.com

Abstract. In this paper, we present a new approach to automatic tagging without requiring any machine learning algorithm or training data. We argue that the critical information required for tagging comes more from word internal structure than from the context and we show how a well designed morphological analyzer can assign correct tags and disambiguate many cases of tag ambiguities too. The crux of the approach is in the very definition of words. While others simply tokenize a given sentence based on spaces and take these tokens to be words, we argue that words need to be motivated from semantic and syntactic considerations, not orthographic conventions. We have worked on Telugu and Kannada languages and in this paper, we take the example of Telugu language and show how high quality tagging can be achieved with a fine grained, hierarchical tag set, carrying not only morpho-syntactic information but also some aspects of lexical and semantic information that is necessary or useful for syntactic parsing. In fact entire corpora can be tagged very fast and with a good degree of guarantee of quality. We give details of our experiments and results obtained. We believe our approach can also be applied to other languages.

Keywords: Tagging, Morphology, Part-Of-Speech, Lexicon, Hierarchical Tag Set, Telugu

1 Introduction

Word classes such as noun, verb, adjective and adverb are called 'Parts of Speech' (POS) by tradition. For the sake of convenience, we may use short labels such as N and V, called tags. Tagging is the process of attaching such short labels to indicate the Parts of Speech for words. One can actually go beyond syntactic categories and/or sub-categories and include lexical, morphological or even semantic information in the tags depending upon the need. In this paper we use the terms Tag and Tagging in this slightly broader sense.

Lexical, morphological and syntactic levels are well recognized in linguistics. Linguistic theories normally do not posit separate tagging or chunking levels at all. There does not seem to be any evidence that the human mind carries out tagging or chunking as separate processes before it embarks upon syntactic analysis.

However, in practice it has generally been found that tagging can significantly reduce lexical ambiguities and thereby speed up syntactic parsing. Tagging is thus useful only to the extent it reduces ambiguities. Of course tagging can also help in other tasks such as word sense disambiguation, text categorization and text summarization.

There are mainly two broad approaches for POS tagging: 1) Linguistic, Knowledge Based or Rule Based approaches 2) Machine Learning or Stochastic or Statistical approaches (HMM and Viterbi decoding, for example). Combinations of the two are also used. We may either do a purely statistical tagging first and then rule out linguistically impossible assignments, or, we may start with linguistically possible tag assignments and then use statistics to the choose the 'best' assignments. Stochastic tagging techniques can be either supervised / unsupervised / hybrid. One may think of tagging as assignment of tags to words or as disambiguation of possible tags. It may be noted that a dictionary or a morphological analyzer typically looks at words in isolation while a tagger looks at the sentential context and attempts to reduce the possible tags for a given word in context in which it appears. Statistical approaches may assign a tag sequence to a word sequence, instead of assigning tags to individual words. Each method has its own merits and demerits.

Machine learning approaches require training data. Generating training data is not an easy task and the quality and quantity may both be important considerations. Training data needs to be large and representative. Labeled training data can be either generated completely manually or tagged data generated by an existing tagger can be manually checked and refined to create high quality training data and both of these methods have their obvious limitations. In practice, we will have to live with sparse data and smoothing techniques used may introduce their own artifacts.

Given the limited amount of training data that is practically possible to develop, a large and detailed tag set will lead to sparsity of training data and machine learning algorithms will fail to learn effectively[1]. Manual tagging and checking also become difficult and error prone as the tag set becomes large and fine-grained and so there is a strong tendency to go for small, flat tag sets in machine learning approaches [2–6]. Such small tag sets may not capture all the required and/or useful bits of information for carrying out syntactic parsing and other relevant tasks in NLP. Morphological features are essential for syntactic analysis in many cases. These have also been the conclusions of a practical experiment of using fine grained morphological tag set reported by Schmid and Laws[7]. Their experiments were carried out using German and Czech as examples of highly inflectional languages. Fine-grained distinctions may actually help to disambiguate other words in the local context. Flat tag sets are also rigid and resist changes. Hierarchical tag sets are more flexible. Thus the design of the tag-set is strongly influenced by the approach taken for tagging. Further, it is also influenced by the particular purpose for which tagging is taken up. A dependency parser of a particular kind may need a somewhat different sort of

sub-categorization compared to, say, parsing using LFG or HPSG. Re-usability of tagged data across applications is an issue.

Although rule based approaches may appear to be formidable to start with, once the proper set of rules has been identified through a thorough linguistic study, there are many things to gain. Linguistic approaches can give us deeper and far-reaching insights into our languages and our mind. Knowledge based approaches generalize well, avoiding over-fitting, errors can be detected and corrected easily, improvements and refinements are easier too. In a pure machine learning approach, we can only hope to improve the performance of the system by generating larger and better training data and re-training the system, whereas in linguistic approaches, we can make corrections to the rules and guarantee the accuracy of tagging. Rule based approaches are also better at guessing and handling unknown words [8].

In this paper, we present an approach that does not depend upon statistical or machine learning techniques and there no need for any training data either. No manual tagging work is involved. We can afford to use a large, fine-grained, hierarchical tag set and still achieve high quality tagging automatically. We get both speed and accuracy. In this paper, we have chosen to render all Telugu words in Roman [9].

2 Previous Work in Indian Languages

English morphology is very simple and direct to implement. Morphological features also very few. The number of tags used for English POS tagging system are not that large: it ranges from 45 to 203 (in the case of CLAWS C8 tag-set) [10]. Also, average number of tags per token is low (2.32 tags per token on the manually tagged part of the Wall Street Journal corpus in the Penn Tree-bank) [11]. The number of potential morphological tags in inflectional rich languages are theoretically unlimited [11]. In English many of the unknown words will be proper nouns but in inflectional and/or agglutinate languages such as Indian languages, many common nouns and verbs may be absent in the training corpus. Therefore, a good morphological analyzer helps [12, 13, 1].

POS tagging for English seems to have reached the top level, but full morphological tagging for inflectionally rich languages such as Romanian, Hungarian, is still an open problem [11]. Indian Languages are highly inflectional and agglutinative too.

A Rule based POS tagger for Telugu has been developed by Center for Applied Linguistics and Translation Studies, University of Hyderabad, India [14]. Here there are 53 tags and 524 rules for POS disambiguation. A Rule based POS tagger for Tamil has been developed by AU-KBC research center, Chennai, India [15]. Here the tag-set developed by IIIT-Hyderabad, consisting of only 26 tags, is used[2]. There are 97 rules of disambiguation. They report a Precision of 92 percent.

Sandipan Dandapat et al proposed a POS tagger for Bangla POS tagging based on Hidden Markov Models (HMM) [16, 17]. The training data set contained

nearly 41,000 words and test data set contained 5,127 words. Further, they made use of semi-supervised learning by augmenting the small labeled training set they had with a larger unlabeled training set of 100,000 words. They have also used a morphological analyzer to handle unknown words. They report an accuracy of around 89% on a test data of 10,000 words.

Pattabhi R K Rao et al. [15] proposed a hybrid POS tagger for Indian languages. Handling of unknown words is based on lexical rules. For Telugu the test data used by them consists of 6,098 words, out of which only 3,547 are correctly tagged. Precision and Recall for Telugu were 58.2% and 58.2% respectively.

Asif Ekbal et al. [18] proposed a HMM based POS tagger for Hindi, Bengali and Telugu. Here they make use of pre-tagged training corpus and HMM. Handling of unknown words is based on suffixes and Named Entity Recognition. Reported accuracies are 90.90% for Bengali, 82.05% for Hindi and only 63.93% for Telugu.

Pranjal Awasthi et al. [19] proposed an approach to POS tagging using a combination of HMM and error driven learning. They have used Conditional Random Fields (CRF), TnT, and TnT with Transformation Based Learning (TBL) approaches and have reported F-measures of 69.4%, 78.94%, and 80.74% respectively for the three approaches for Hindi.

Sankaran Baskaran [20] used HMM based approach for tagging and chunking. He achieved a Precision of 76.49% for tagging and 55.54% for chunking using the tag-set developed in IIIT-Hyderabad [2], consisting of only 26 tags.

Himanshu Agrawal and Anirudh Mani [21] presented a CRF based POS tagger and chunker for Hindi. Various experiments were carried out with various sets and combinations of features which mark a gradual increase in the performance of the system. A morph analyzer was used to provide extra information such as root word and possible POS tags for training. Training on 21,000 words, they could achieve an accuracy of 82.67%.

Thus, most of the work done so far report accuracies of up to about 90% when tagged with small, flat tag sets. As we shall see, our approach guarantees much higher accuracies although we use a very large, fine grained, hierarchical tag set. Unlike other systems reported above, our system has been tested on very large data.

3 Morphology Based Tagging

The main difference between our approach and all other work on tagging, whether for Indian languages or for other languages of the world, is the way we define words. The general practice is to tokenize sentences based on spaces and take for granted that these tokens are words. Sequences of characters separated by spaces are not necessarily proper linguistic units. Words have to be defined based on meaning and morphological and syntactic properties. We define a word as a sequence of phonemes bearing a definite meaning and having certain syntactic relations with other words in the given sentence. We need to define a set of syntactic relations that are universally applicable to all human languages. For

example, a word which indicates an activity is a verb. If there is one activity, there can be only one verb. Thus 'has been running' is one word, not three. Similarly, 'from the book' is one single word - prepositions, post-positions are not universal word classes, 'from' is not a word in itself, it only adds a morpho-syntactic feature to 'book'. Viewed from this perspective, English morphology is not significantly simpler than the morphology of any other language. Thus, although 'book' and 'books' are both ambiguous between a noun and a verb in English, the words 'from the book' and 'from the books' are both unambiguous and it is morphology which is disambiguating here. This theory of words is a very significant research contribution to NLP and modern linguistics and full details are published elsewhere [22, 23].

Statistical approaches assume that the information necessary for tag assignment comes from the other tokens in the sentence. In many cases, only the tokens that come before the current word are taken into direct consideration. We believe, in sharp contrast, that the crucial information required for assigning the correct tag comes from within the word, in all languages of the world. The crux of tagging lies in morphology. This is clearly true in the case of so called morphologically rich languages but we believe this is actually true of all human languages if only we define words properly, in terms of meanings and universal grammatical properties, rather than in terms of the written form as a sequence of characters delimited by spaces.

A vast majority of the words can be tagged correctly by looking at the internal structure of the word. In those cases where morphology assigns more than one possible tag, information required for disambiguation comes mainly from syntax. Syntax implies complex inter-relationships between words and looking at a sentence as a mere sequence of words is not sufficient. Statistical techniques are perhaps not the best means to capture and utilize complex functional dependencies between words in a sentence. Instead, syntactic parsing will automatically remove most of the tag ambiguities. It must be reiterated that tagging is intended only to reduce tag ambiguities, not necessarily to eliminate all ambiguities. Syntactic parsing systems are anyway capable of handling ambiguities.

Identifying words is thus a critical task, mere tokenization based on white spaces will not do. In Dravidian languages (including Telugu, Kannada, etc.), as also in Sanskrit, the difference between orthographic tokens and proper words is not too much. Whatever be the case, differences can be handled using several techniques. A pre-processing module can be introduced with the main intention of first tokenizing and then obtaining words from these tokens. In Telugu, we do this using regular expression based pattern matching rules. Languages like English and Hindi may require more complex rules. In certain cases, mainly sandhi (phonetic conflation) and compounds, the morphology module is itself designed to handle these differences. A post-morphology bridge module ensures that we finally have proper words, tagged and ready for further processing such as syntactic parsing.

The lexicon assigns tags to words that appear without any overt morphological inflection. Morphology handles all the derived and inflected words, including

many forms of sandhi. The bridge module combines the tags given by the dictionary and the additional information given by the morph, ensuring that the correct structure (and hence meaning) are depicted by the tags. The overall tag structure remains the same throughout, making it so much simpler and easier to build, test and use.

The morph system is implemented as an extended Finite State Transducer. The FST has 398 transitions or arcs. The figure below shows a small part of the FST. A category field has been incorporated so that only relevant transitions are allowed. Derivation is handled by allowing category changes. Transitions are on morphemes, not on individual characters or letters. Dravidian morphology involves complex morpho-phonemic changes at the juncture of morphemes and linguistically motivated rules have been used to handle these [24].

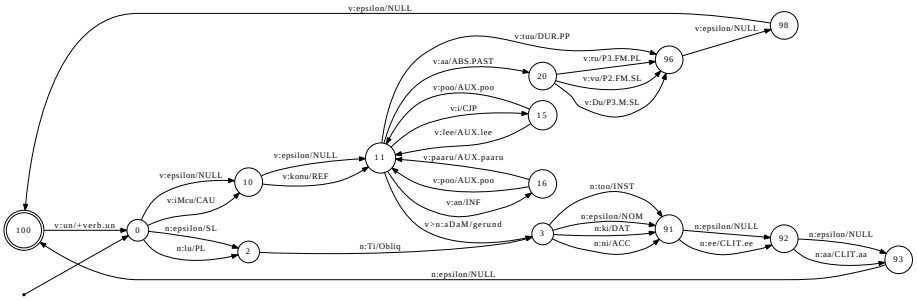


Fig. 1. Sample FST Grammar

We find that in any running text approximately 40% of the words are found directly in the dictionary. Less than 2% of the words in the dictionary are ambiguous. About one third of these are ambiguous between noun and verb. Since nominal and verbal morphologies are more or less completely disjoint in Telugu, and since these words occur mostly in inflected forms (more than 92% of times), morphology can resolve most of these cases of ambiguity. Morphology can also resolve ambiguity between nouns / verbs other categories such as adjectives and adverbs. Thus, morphology has a very important role in tagging. If we work with proper words instead of tokens, we believe we will get a similar picture in other languages. Certain kinds of systematic structural ambiguities in a language can lead to multiple tag assignments, calling for further disambiguation.

4 Tag Set Design and Tagging

Tags must be assigned to words, not to tokens. This is where we differ from all others. Once we have a precise definition of what constitutes a word and once we have a clear idea of universal word classes, the main grammatical categories and tags can be defined accordingly. The main categories should ideally be semantically motivated and hence universal and language independent. Nouns and

verbs are universal categories with an independent and clear lexical meaning. Adjectives and manner adverbs have dependent lexical meaning and can also be taken as universal categories. Pronouns are variables, they do not have a fixed lexical meaning, but their meaning can be resolved in context. These five are the universal lexical categories. Conjunctions are typical of functional categories.

Although the major categories are semantically motivated, it must be noted that in the actual analysis process, we start from characters, build tokens and hence words, and work bottom-up through dictionary look-up / morphological analysis towards syntactic analysis leading to semantics. Since computers cannot work directly with meanings, we will have to work keeping lexical, morphological and syntactic properties in mind. Subcategories are thus dependent to some extent on the intended purpose and architectural and design issues. Each tag should then be precisely defined and supported with examples, need and justification. We give here the summary of our tagging scheme - see [25] for more details.

Table 1: LERC-UoH Tag Set

N (NOUN)		ADV (Adverbs)	
	COM(Common)		MAN(Manner)
	PRP(Proper)		CONJ(Conjunctive)
	-PER(Personal)		PLA(Place)
	-LOC(Location)		TIM(Time)
	-ORG(Orgzn.)		NEG(Negative)
	-OTH(Others)		QW(Question Word)
	LOC(Locative)		INTF(Intensifier)
	CARD(Cardinal)		POSN(Post-Nominal
PRO (Pronoun)			Modifier)
	PER(Personal)		ABS (Absolute)
	INTG(Interrogative)	CONJ (Conjunction)	
	REF(Reflexive)		SUB(Subordinating)
	INDF(Indefinite)		COOR(Coordinating)
ADJ (Adjective)		V (Verb)	
	DEM(Demonstrative)		IN(Intransitive)
	QNTF(Quantifying)		TR(Transitive)
	ORD(Ordinal)		BI(Bitransitive)
	ABS(Absolute)		DEFE(defective)
SYMB (Symbol)		INTJ (Interjection)	

Here are some examples of tags in the dictionary.

```
baDi||N-COM-COU-N.SL-NOM
aMdamaina||ADJ-ABS
adhikaari||N-COM-COU-FM.SL-NOM
ataDu||PRO-PER-P3.M.SL-DIST-NOM
muduru||ADJ-ABS||V-IN
telusu||V-DEFE
tinu||V-TR
paatika||N-CARD-NHU-NOM
```

Here PRO-PER-P3.M.SL-DIST-NOM as a whole is called a tag. A tag consists of a series of tag elements separated by hyphens. The first element is always

the main category and the next one or two levels indicate syntactic or morphological subcategories. The rest are morphological or semantic features. There is a more or less one-to-one correspondence between these elements and the morpheme structure of words. When a morpheme indicates more than one feature, the individual features are indicated as tag atoms within the given element, as in the case of P3.N.SL. In our Telugu dictionary, there are 274 unique tags made up of 143 tag elements and 121 atoms. Morph refines and/or adds more information. For example, 'ceppu' is a verbal root listed in the dictionary and 'ceppinavaaDu' is a pronominalized form derived by morphology and the corresponding tags are:

```
ceppu||N-COM-COU-N.SL-NOM||V-TR12

ceppinavaaDu||ceppu||V-TR12.v-PAST.RP-.adj
                     -PRON.vaaDu.P3.M.SL-.n-NOM
```

In the final analysis there are more than 20,000 tags for nouns (including number, case, clitics, vocatives, pronominalized forms, etc.) and nearly 15 Million different tags for verbs (including inflection, derivation, clitics etc.) Our morph is capable of generating and analyzing all these word forms. The tags contain all the necessary lexical, morphological, syntactic and relevant semantic information for carrying out syntactic analysis etc. without need for getting back to the dictionary or morphology.

Most of the other works on morphology for Indian languages are based on the Paradigm Model where lists of word forms are manually created for each paradigm based on morpho-phonemic considerations but as reflected in the orthography. It is next to impossible to create complete lists of all word forms manually given the richness of morphology of our languages. Nor is this an intelligent or wise approach. It is very unlikely that the human mind simply lists all forms of all words in tables. Also, morphology is reduced to arbitrary string manipulation in this paradigm approach. For example, in Telugu, 'maniShi' (person) becomes 'manuShulu' (persons) in plural. In the paradigm approach, 'man' is identified as the common prefix and 'maniShi' is broken into 'man' and 'iShi'. Then, 'manuShulu' is obtained by adding 'uShulu' to 'man'. Since 'man', 'iShi', 'uShulu' are all totally arbitrary, meaningless, linguistically unacceptable units, this is really not morphology at all. Ours is perhaps the first, linguistically motivated, psychologically plausible, nearly complete, computationally efficient morphological system for any Indian language. It may be noted that many other works for various languages across the world are also based on arbitrary character level manipulations. A proper system of morphology will be of great help not only in tagging but also for spell checking, stemming / lemmatization etc. More importantly, it will provide insights into the way the language works. A proper system of morphology will be useful for language teaching and learning too.

Morph can resolve a major portion of tag ambiguities. For example, the Telugu word 'ceppu' has two meanings: 1) 'to say or to tell' 2) shoe or slipper.

The examples below show how morphology can resolve the noun-verb ambiguity. In the case of derivations, note how our tags depict the complete flow of category changes. This is essential for syntactic parsing.

```
ceppu||N-COM-COU-N.SL-NOM||V-TR12

ceppaaDu||ceppu||V-TR12-ABS.PAST-P3.M.SL

ceppinavaaDu||ceppu||V-TR12.v-PAST.RP-.adj
                     -PRON.vaaDu.P3.M.SL-.n-NOM

ceppulanu||ceppu||N-COM-COU-N.PL-ACC
```

When morph fails to disambiguate, syntactic considerations such as chunking constraints, predicate-argument structure and selectional restrictions can resolve the ambiguities in most cases. Less than 1% of words will remain ambiguous as can be seen from our experiments below.

Disambiguation by purely statistical methods have also been used by researchers [26]. Although all words can be disambiguated, there can be no guarantee of correctness, even in cases where clear disambiguation rules exist linguistically. A rule-based disambiguation will usually leave out only those ambiguities which are genuine.

5 Experiments and Results

There are no publicly available standard data sets available for Telugu. We have developed our own Telugu text corpus of about 50 Million words [27]. We have tested our system on a corpus of 15 Million words. Performance of the morph analyzer on randomly selected sentences from this corpus is shown below:

Table 2. Results of Morph Analysis on Telugu Corpora

#Sent	#Tokens	Found in Dict	Identified by Morph	Unknown
101	861	(376) 44%	(402) 46%	(83) 10%
500	4788	(2058) 43%	(2330) 49%	(400) 8%
1000	9269	(3869) 42%	(4691) 50%	(709) 8%
1500	14092	(5860) 42%	(7105) 50%	(1127) 8%

Eight to ten percent of the words remain un-analyzed. We have options for guessing but here we show results without guessing. It is found on close inspection that most of the un-analyzed words are spelling errors, loan words, named entities and compounds. Among the words analyzed, it is found that around

10% of words are assigned more than one tag. In most cases of ambiguity, words get only two tags, not more. More importantly, the correct tag is almost always included.

Since ours is a manually created rule based system, there is no scope for chance errors. Incorrect analysis is very rare and occurs only due to complex interactions involving spelling errors, loan words, named entities etc. In order to evaluate the Precision and Recall, random samples have been manually checked. A random sample of 202 sentences consisting of 1776 words has been tagged and manually checked carefully. Of these, 1626 words (91.5%) were tagged, the rest remain untagged. Only 5 words (0.3%) were found to be incorrectly tagged. This gives us a Precision of 99.69% and a Recall of 91.27%. In these calculations, a word has been taken to be correctly tagged if the correct tag is included, along with possibly other tags.

In cases of ambiguous tag assignments, we use a set of 17 rules based on local syntactic context to disambiguate the tags. About 90% of ambiguities can be resolved using these local rules. Finally, we find that we can tag more than 93% of all words in a raw corpus, with less than 1% of the words assigned more than one tag, and with a guarantee of more than 99% correctness.

6 Conclusions

In this paper we have presented a new approach to tagging based on our new theory of words, using a morphological analyzer and a fine-grained hierarchical tag-set. We have shown that it is possible to develop high performance tagging system without need for any training data or machine learning or statistical inference. Since the whole system is rule governed, the results can be guaranteed to be correct. Manual verification has validated this claim. We have demonstrated the viability and merits of our ideas through actually developed system for Telugu. The same ideas and methods have been used to develop a system for Kannada and the performance of our Kannada system is similar. The method is being applied for other languages too.

References

1. Atwell, E.: Development of Tag Sets for Part-of-Speech Tagging. In Ludeling, A., Kyto, M., eds.: Corpus Linguistics An International Handbook, Mouton de Gruyter (2008) 501–526
2. IIIT-Hyderabad: A Part-of-Speech Tagset for Indian Languages. `http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf`
3. AU-KBC: POS Tagset for Tamil. `http://nrcfosshelpline.in/smedia/images/downloads/Tamil_Tagset-opensource.odt`
4. Sankaran, B., Bali, K., Choudhury, M., Bhattacharya, T., Bhattacharyya, P., Jha, G., Rajendran, S., Saravanan, K., Sobha, L., Subbarao, K.V.: A Common Parts-of-Speech Tagset Framework for Indian Languages. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, European Language Resources Association (ELRA) (2008) 1331–1337

5. RamaSree, R.J., Rao, G.U., Murthy, K.V.M.: Assessment and Development of POS Tagset for Telugu. In: Proceedings of the Sixth Workshop on Asian Language Resources, 3rd International Joint Conference on Natural Language Processing (IJCNLP-08), IIIT Hyderabad, Hyderabad, India (2008) 85–88
6. Elworthy, D.: Tagset Design and Inflected Languages. In: In EACL SIGDAT workshop From Texts to Tags: Issues in Multilingual Language Analysis. (1995) 1–10
7. Schmid, H., Florian, L.: Estimation of Conditional Probabilities With Decision Trees and an Application to Fine-Grained POS Tagging. In: COLING. (2008) 777–784
8. Abney, S.: Part-of-Speech Tagging and Partial Parsing. In: Corpus-Based Methods in Language and Speech, Kluwer Academic Publishers (1996) 118–136
9. Murthy, K.N., Srinivasu, B.: Roman Transliteration of Indic Scripts. In: 10th International Conference on Computer Applications, University of Computer Studies, Yangon, Myanmar (28-29 February 2012)
10. Garside, R.: The CLAWS Word-Tagging System. In Garside, R., Leech, G., Sampson, G., eds.: The Computational Analysis of English, Longman (1987) 30–41
11. Hajič, J.: Morphological Tagging: Data vs. Dictionaries. In: Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference, Seattle, Washington (2000) 94–101
12. Huihsin, T., Jurafsky, D., Christopher, M.: Morphological Features help POS Tagging of unknown Words across Language Varieties. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, Association for Computational Linguistics (October 2005) 32–39
13. Sawalha, M., Atwell, E.: Fine-grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta (2010) 1258–1265
14. SreeGanesh, T.: Telugu POS Tagging in WSD. In Journal of Language in India **6** (August 2006)
15. Pattabhi, R.K.R., SundarRam, R.V., Krishna, R.V., Sobha, L.: A Text Chunker and Hybrid POS Tagger for Indian Languages. In: Proceedings of International Joint Conference on Artificial Intelligence Workshop on Shallow Parsing for South Asian Languages, IIIT Hyderabad, Hyderabad, India (2007)
16. Dandapat, S., Sarkar, S., Basu, A.: A Hybrid Model for Part of Speech Tagging and its Application to Bengali. In: Proceedings of International Conference on Computational Intelligence, Instanbul, Turkey (2004) 169–172
17. Dandapat, S., Sarkar, S.: Part of Speech Tagging for Bengali with Hidden Markov Model. In: Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT Hyderabad, Hyderabad, India (2006)
18. Ekbal, A., Mandal, S.: POS Tagging using HMM and Rule based Chunking. In: Proceedings of International Joint Conference on Artificial Intelligence Workshop on Shallow Parsing for South Asian Languages, IIIT Hyderabad, Hyderabad, India (2007)
19. Awasthi, P., DelipRao, Ravindran, B.: Part of Speech Tagging and Chunking with HMM and CRF. In: Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT Hyderabad, Hyderabad, India (2006)

20. Baskaran, S.: Hindi Part of Speech Tagging and Chunking. In: Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT Hyderabad, Hyderabad, India (2006)
21. Agarwal, H., Mani, A.: Part of Speech Tagging and Chunking with Conditional Random Fields. In: Proceedings of NLPAI Machine Learning Workshop on Part of Speech Tagging and Chunking for Indian languages, IIIT Hyderabad, Hyderabad, India (2006)
22. Murthy, K.N.: Language, Grammar and Computation. Central Institute of Indian Languages (CIIL), Mysore (Forthcoming)
23. Murthy, K.N.: What Exactly is a Word? Special Issue of International Journal of Dravidian Language (Forthcoming)
24. Krishnamurthi, B., Gwynn, J.P.L.: A Grammar of Modern Telugu. Oxford University Press, New Delhi (1985)
25. Murthy, K.N., Srinivasu, B.: On the Design of a Tag Set for Dravidian Languages. In: 40th All India Conference of Dravidian Linguists, University of Hyderabad, Hyderabad, India (18-20 JUNE 2012)
26. Steven, J., DeRose: Grammatical Category Disambiguation by Statistical Optimization. Computational Linguistics **14**(1) (1988) 31–39
27. Kumar, G.B., Murthy, K.N., Chaudhuri, B.B.: Statistical Analysis of Telugu Text Corpora. In International Journal of Dravidian Languages **36**(2) (June 2007) 71–99

CPRel: Semantic Relatedness Computation Using Wikipedia based Context Profiles

Shahida Jabeen, Xiaoying Gao and Peter Andreae

School of Engineering and Computer Science
Victoria University of Wellington, New Zealand
[shahidarao, Xiaoying.Gao, Peter.Andreae]@ecs.vuw.ac.nz

Abstract. Semantic relatedness is a well known problem with its significance ranging from computational linguistics to Natural language Processing applications. Relatedness computation is restricted by the amount of common sense and background knowledge required to relate any two terms. This paper proposes a novel model of relatedness using context profile built on features extracted from encyclopedic knowledge. Proposed research makes use of Wikipedia to represent the context of a word in the high dimensional space of Wikipedia labels. Semantic relatedness of a word pair is then assessed by comparing their corresponding context profiles based on three different weighting schemes using traditional Cosine similarity metrics. To evaluate proposed relatedness approach, three well known benchmark datasets are used and it is shown that Wikipedia article contents can be used effectively to compute term relatedness. The experiments demonstrate that the proposed approach is computationally cheap as well as effective when correlated with human judgments.

1 Introduction

Semantic relatedness is the process of quantifying the extent of semantic connection between two textual units [1,2]. Semantic relatedness is a well explored area. Many researchers have attempted to solve this problem by taking into account various aspects such as statistical relatedness, lexical relations, text contents, rhetorical relations and using external sources of world knowledge such as thesaurus, lexical databases, dictionaries and encyclopedia. Consequently, this problem is widely studied in a variety of applications ranging from computational linguistics to NLP and web mining to intelligent web. There are various applications of semantic relatedness in text summarization [3], information retrieval [4], topic identification [5,6], automatic keyphrase extraction [7], topic indexing [8], word sense disambiguation [9,10,11], document clustering [12,5] and spelling correction [13].

Traditional way of computing text relatedness is to represent context of individual words in a multidimensional space and computing the distance between their corresponding vectors. This paper introduces a new model of relatedness called *Context Profile based Relatedness (CPRel)*. CPRel improved the context

representation by constructing the context profile of each concept based on certain features derived from Wikipedia. The proposed research focuses on computing semantic relatedness of individual words. However, it can be conveniently used for text relatedness as well. It should also be noted that this work focuses on semantic relatedness computation rather then semantic similarity which is less generalized and is based on specific lexical relations such as synonymy, hypernymy or hyponymy. In all of the vector space inspired approaches, the selection of the high-dimensional context space plays a vital role in controlling the performance of relatedness computation. Proposed research analyzed this aspect of relatedness and with the help of simple features achieved a performance comparable to other well known Wikipedia based approaches.

The rest of the paper is organized as follows. Section 2 categorizes existing semantic measures proposed in literature and the corresponding research done in each category. Section 3 proposes and discusses a new relatedness approach. Section 4 analyzes the performance of proposed methodology using three well known datasets. Comparison of proposed approach with other existing strategies and discussion on results are also included in the same section. Finally, section 5 concludes this research and discusses some future research directions.

2 Related Work

With the exponential growth of World Wide Web and ever increasing importance of retrieving relevant information from web, contextual relatedness has become a critical research area. Prior work on relatedness computation can be divided into two main streams: *Statistical techniques based approaches*, where text content and corpus features are statistically analyzed to compute relatedness scores and *external knowledge source based approaches*, where repositories of human knowledge are used as a source of background knowledge to support relatedness computation.

Early research work based on Statistical techniques, introduced the concept of distributional similarity[14,15]. Later, Latent Semantic Analysis (LSA) [16] was proposed as a dimensionality reduction technique where latent concepts are represented by most prominent dimensions in the data using Singular Value Decomposition (SVD). Similarly, Hoffman proposed Probabilistic LSA [17] that constructs a low dimensional concept space. Another statistical technique used for relatedness computation is Latent Dirichlet Allocation (LDA)[18]. LDA represents a document as a mixture of words where each word is attributable to one of the document topics. Sun et al. [19] used LDA based Fisher Kernel for text segmentation.

Various attempts were made to incorporate human knowledge in a structured way in relatedness computation using external knowledge sources such as knowledge bases, dictionaries, thesauri and lexical databases. Ponzetto and Strube [2] used Wikipedia category network and calculated various statistical and structural measures from Wikipedia concepts. Yeh et al. [20] constructed Wikipedia graph and applied random walk with personalized page ranks to compute se-

mantic relatedness for words and texts. Gabrilovich and Markovich [21] proposed Explicit Semantic Analysis (ESA) to incorporate human knowledge into relatedness computation by constructing concept vectors and comparing them using Cosine Similarity. Milne and Witten [22] used Wikipedia hyperlink structure to compute semantic relatedness based on in-link and out-link overlaps. Temporal Semantic Analysis (TSA) [23] was proposed to incorporate temporal dynamics to enhance text relatedness models. TSA represented each input word as a concept vector and extended static representation with temporal dynamics. Jabeen et al. [24] used Wikipedia hyperlinks and disambiguation pages for relatedness computation. They used Dice Coefficient inspired measure of relatedness. Halawi et al. [25] proposed Constrained Learning of relatedness in which they learned a suitable word representation in a latent factor space. Hassan and Mihalceae [1] introduced Salient Semantic Analysis (SSA) by modeling frequently co-occurring words in a contextualized profile for each word. They only used words with high saliency or relevance to the document. Their approach works for both word pairs and text pair relatedness computation. Liu et al. [26] incorporated UMLS and WordNet definitions to generate context vectors for relatedness computation on biomedical data. Navigli and Ponzetto [27] proposed a graph based multilingual approach to compute semantic relatedness. They used BabelNet, a multilingual lexical knowledge source, to construct sub graphs for a word pair in different languages and computed semantic relatedness based on the subgraph intersection.

Proposed approach effectively bridges previously mentioned two research streams by augmenting the relatedness computation with statistics derived from Wikipedia as an external knowledge source. Frequency of occurrence and link probability are used as statistical features driven from Wikipedia article contents. Proposed research is similar to ESA with three main distinctions: First, proposed approach considers a different context for each input word, based on Wikipedia article contents. Second, proposed approach is computationally cheap as it does not preprocess the entire Wikipedia like ESA and third, proposed approach does label pruning to filter out unwanted context and showed that this technique is quite effective in improving the performance of proposed relatedness measure.

3 Context Profile based Relatedness Computation

The idea behind the proposed relatedness computation method is to construct a context profile of each input word based on the corresponding Wikipedia article. Label pruning is performed to weed out all unnecessary labels. In the context profile, each label is assigned a weight based on a hybrid weighting scheme. Semantic relatedness of word pair is then assessed by comparing their corresponding context profiles using traditional Cosine similarity metrics. The work flow of the proposed method is shown in Figure. 3.

To compute the relatedness score of two given terms, the first step is to identify their corresponding Wikipedia articles. The shear size of Wikipedia is

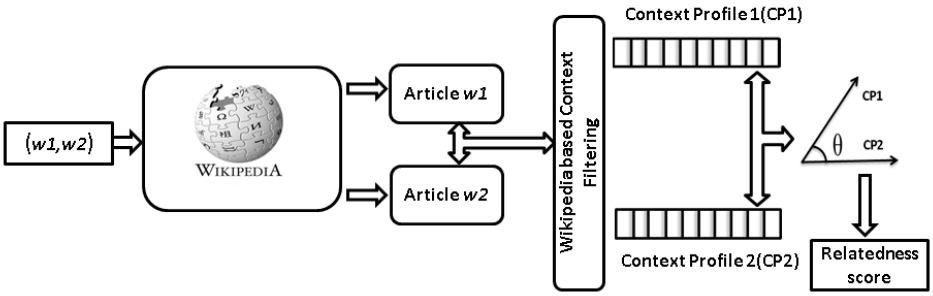


Fig. 1. Framework for Context Profile based Relatedness Computation

big enough to cope with the knowledge requirement of all datasets used in this research.

A series of preprocessing steps is followed to convert Wikipedia articles from *MediaWiki* format to plain text. These steps involve stripping external links, non-article links, special characters and extra spaces. After the article matching and preprocessing phase, article contents are gone through a number of filtering steps to eliminate unnecessary words from the context profile of each word.

Classical ways of text filtering is to weed out all stop words for which the list of common English words[1] is used. Stemming is also performed to convert inflectional words to their roots. N-grams up to 3-grams are extracted from article contents. Clearly, many n-grams would be of no help in supporting the context of a specific term. To prune such words, Wikipedia labels are used. In Wikipedia anchor texts, also called labels, are the hypertexts used to link one Wikipedia article to other articles within the same context. This way all the Wikipedia articles are linked to other articles through the hyperlink structure, making Wikipedia a graph of links. These anchor texts are a very good source of encoding synonyms and other variations in the title of an article. They are an additional source of finding those synonyms which are not covered by the redirects. They are an extremely useful component of Wikipedia because Wikipedia contributors modify them according to the context of the article in which they are used. They not only encode the synonyms and surface forms but also the polysemy and the likeliness of each sense [28]. Matching with Wikipedia labels is a good way of judging whether a word or phrase is useful or not. *Link Probability* (LP) [11] is a proven measure to signify keyphraseness of a word. This research uses Wikipedia labels and link probability for context filtering. filtering of unwanted context is performed in two phases. In the first filtering pass, all words which are not valid Wikipedia labels are discarded, leaving only those keywords that which match with Wikipedia labels. In the second pass, all labels having

[1] available at http://www.db-net.aueb.gr/gbt/resources/stopwords.txt

LP values below a certain cutoff threshold α are discarded. So, $k \epsilon CP(w1)$ if $LP(k) > \alpha$ where, $CP(w1)$ is the context profile of input word $w1$ and $LP(k)$ represents the LP value of a label k. Labels with LP value above threshold α are used to populate the context profile of each input word. This way context profile of each input word is represented in high dimensional space of Wikipedia labels.

3.1 Normalized Term frequency and Link Estimation

Each label in the context profile of each input word is assigned a hybrid weight based on two features:

- **Term frequency**: If a word occurs in a good proportion to the total size of a specific article then it is considered important for that article. Based on this assumption, Term Frequency (NTF) of a word w is computed as the number of times w occurs in a specific Wikipedia article normalized by the size of that article and is given by:

$$TF(w) = \frac{Count(w)}{|W|} \tag{1}$$

  Where $|W|$ is the total number of words in the article.

- **Link probability**: If a keyword occurs more number of times as a label in Wikipedia then it is significant. Based on this assumption, Link probability is used to signify the importance of a keyword as a label. It is the defined as an estimation of probability of a keyword to be used as a link in Wikipedia. It is defined as a ratio of the number of Wikipedia documents having a keyword as a link to the number of Wikipedia documents in which that keyword occurs in any form (as a link or a word) and is given by:

$$P(keyword|w) = \frac{count(Dkey)}{count(Dw)} \tag{2}$$

Where, $count(Dkey)$ represents the number of documents having a word w as an label and $count(Dw)$ is the number of documents in which the word appears. In general, the more generic a Wikipedia label is, the less is its link probability. So a label *Car* gets a lower LP value(0.01) than a label *Sports Car*, which gets a lower Lp value (0.17) than *Ferrari* with LP value 0.27, whereas, most generic labels such as *the* gets extremely lower LP value (8.7×10^{-6}).

These two features are modified and combined to assign weights to individual labels of context profiles. So, after stemming, each root word r is assigned a weight w based on individual weights of its inflectional words set [*w1,w2....wn*].

$$w(r) = LE(r) \times NTF(r) \tag{3}$$

where, *Normalized Term Frequency (NTF)* is defined as the sum of frequencies of all the inflectional words divided by total number of words in an article and is given as:

$$NTF(r) = \frac{\sum_{i=1}^{k} TF(w_i)}{|W|} \tag{4}$$

In general, *NTF* is good for finding out frequently occurring relevant words in a document but its not always helpful. Some of the labels may still exist in an article with high frequency count but of not much relevance. To counter such keywords, *Link Estimation (LE)* of a root word r is used as the measure of popularity of a root word being used as a link in the whole corpus. It is defined as the ratio of sum of link document count (Number of documents where the word occur as a link) of each inflectional form to the sum of total document count (Number of document where a word occurs at all) of them. *LE* is computed as below:

$$LE(r) = \frac{\sum_{i=1}^{k} count(Dkey_{w_i})}{\sum_{i=1}^{k} count(Dw_{w_i})} \tag{5}$$

where, k represents the number of inflectional forms of a root word r. This measure penalizes all unwanted common words which succeeded in passing through stop word and label filters and have higher *NTF*.

4 Evaluation

For relatedness computation based on proposed approach, the version of Wikipedia released in July 2011 is used. At this point, it contains 33GB of uncompressed XML markup which corresponds to more than five million articles, sufficiently covering all concepts for which manual judgment are available.

According to Budanitsky and Hirst [29,30], there are three methods of semantic relatedness evaluation: *Mathematical analysis*, where formal properties of relatedness measure are assessed, *application specific evaluation*, where the measure is applied in a real world application and tested indirectly and *comparison with human judgment*, where human judgments are used as gold standard for evaluation. Third method is the most widely used and best suited application independent evaluation method for relatedness computation. Proposed research also followed the same method for evaluation of relatedness computation.

In this experiment, three standard datasets, which have been widely used in the existing relatedness research, are used:

R&G dataset: Rubenstein and Goodenough (R&G) dataset consists of 65 words pairs sorted in an increasing order of relatedness. These 65 word pairs are scored by 51 human judges on a scale of 0-4 where 0 means unrelated and 4 means exactly the same.

M&C dataset: Miller and Charles dataset is a noun subset of R&G dataset and consists of 30 word pairs which are scored by 38 human subjects on the scale of 0-4.

WordSimilarity-353: WordSimilarity-353 also known as Finkelstein-353 is a dataset of 353 word pairs scored by 13 human experts on a scale of 0-10. It also includes 30 word pairs of M&C dataset but unlike M&C it includes diverse range of word pairs from proper nouns like "Yasser Arafat" to phrases like "Wednesday News" and abbreviations like "FBI" and "OPEC" which adds extra difficulty to the relatedness measure evaluation.

Many word pairs in these sets include ambiguous word like (*Crane,tool*). Since disambiguation is beyond the scope of this research so manually disambiguated versions of M&C and WordSimilarity-353[2]datasets were used and R&G dataset was manually disambiguated based on Wikipedia articles. Some of word pairs in these datasets do not have corresponding Wikipedia articles so such word pairs were also removed from each dataset, resulting in 24 word pairs in M&C, 58 word pairs in R&G dataset and 314 pairs in WordSimilarity-353 dataset.

Table 1. Best performance of three variants of CPRel on three benchmark datasets

Dataset	CPRel(hybrid)		CPRel(NTF)		CPRel(LE)	
	r_s	r	r_s	r	r_s	r
M&C	0.83	0.70	0.83	0.79	0.81	0.66
R&G	0.79	0.66	0.79	0.71	0.79	0.63
WordSimilarity-353	0.66	0.43	0.69	0.52	0.64	0.53

Two other variants of proposed relatedness measure are used to compare and analyze the influence of individual and combined features on the relatedness computation: CPRel with NTF weighting scheme and CPRel with LE weighting scheme. The comparison of CPRel with three different weighting schemes using Spearman's Correlation Coefficient (r_s) and Pearson Correlation (r) is shown in Table. 1. In general, the performance of all three variants of CPRel is highest on M&C dataset using both correlation variables whereas, both CPRel (Hybrid) as well as CPRel (NTF) achieved highest correlation values on M&C dataset overall. This solidify the fact that the proposed approach works quite well on noun-noun word pairs since M&C is a noun subset of R&G dataset.

One of the main features of proposed method is context filtering based on LP cutoff threshold. Performance of the proposed system varies according to the chosen cutoff threshold value. To understand the impact of label pruning on improving relatedness, the behavior of proposed relatedness approach on various cutoff threshold values is tested. For each dataset, six different threshold values (between 0 and 1) were chosen randomly and both Spearman's Correlation and Pearson's Correlation values were computed for CPRel with three different weighting schemes. The effect of cutoff threshold on each of the dataset is shown

[2] The manually disambiguated WordSimilarity-353 dataset is available at: http://www.nzdl.org/wikipediaSimilarity

in Table. 5. Only the best and worst r_s values of each variant of CPRel are shown[3].

Table 2. Performance variations of three versions of CPRel on M&C dataset

LP Cutoff Threshold	CPRel(hybrid)		CPRel(NTF)		CPRel(LE)	
	r_s	r	r_s	r	r_s	r
0.001	0.83	0.70	0.80	0.75	0.81	0.66
0.005	0.76	0.70	0.83	0.79	0.81	0.66
0.01	0.81	0.70	0.77	0.74	0.81	0.66
0.05	0.76	0.70	0.74	0.69	0.81	0.66
0.1	0.79	0.64	0.76	0.74	0.81	0.66
0.5	0.68	0.70	0.69	0.65	0.70	0.65

Table 3. Performance variations of three versions of CPRel on R&G dataset

LP Cutoff Threshold	CPRel(hybrid)		CPRel(NTF)		CPRel(LE)	
	r_s	r	r_s	r	r_s	r
0.001	0.78	0.66	0.67	0.70	0.79	0.63
0.005	0.78	0.66	0.64	0.71	0.79	0.63
0.01	0.78	0.66	0.76	0.70	0.78	0.63
0.05	0.76	0.65	0.78	0.66	0.78	0.63
0.1	0.71	0.66	0.70	0.66	0.77	0.63
0.5	0.79	0.65	0.79	0.65	0.79	0.63

Table 4. Performance variations of three versions of CPRel on WS-353 dataset

LP Cutoff Threshold	CPRel(hybrid)		CPRel(NTF)		CPRel(LE)	
	r_s	r	r_s	r	r_s	r
0.001	0.65	0.43	0.62	0.47	0.61	0.53
0.005	0.65	0.43	0.59	0.51	0.60	0.53
0.01	0.66	0.43	0.69	0.51	0.64	0.53
0.05	0.60	0.42	0.61	0.52	0.60	0.53
0.1	0.62	0.41	0.58	0.35	0.58	0.51
0.5	0.61	0.42	0.59	0.48	0.59	0.50

In general, high cutoff value means more keywords are filtered out in the context pruning phase. The effect of LP cutoff threshold on performance of CPRel using M&C dataset is shown in Table. 2. Clearly, with all three variants of

[3] Bold values in Table. 5 indicate a specific performance pattern of CPRel. Clearly, on M&C dataset the performance of all variants of CPRel was consistently worst on LP cutoff threshold value of 0.5 whereas for R&G dataset, the best correlation values were achieved on α=0.5. In case of WS-353, the best performance was observed on α=0.01

Table 5. Correlation based Performance of three weighting schemes of CPRel on three benchmark datasets

Dataset	CPRel(hybrid)			CPRel(NTF)			CPRel(LE)		
	Result	Cutoff	r_s	Result	Cutoff	r_s	Result	Cutoff	r_s
M&C	Best	0.001	0.83	Best	0.005	0.83	Best	0.001	0.81
	Worst	**0.5**	0.68	Worst	**0.5**	0.69	Worst	**0.5**	0.70
R&G	Best	**0.5**	0.79	Best	**0.5**	0.79	Best	**0.5**	0.79
	Worst	0.1	0.71	Worst	0.005	0.64	Worst	0.1	0.77
WS-353	Best	**0.01**	0.66	Best	**0.01**	0.69	Best	**0.01**	0.64
	Worst	0.05	0.60	Worst	0.1	0.58	Worst	0.1	0.58

CPRel on M&C datset, the highest correlation value is achieved on lowest cutoff value and vice versa. But, in case of R&G dataset, shown in Table. 3, the behavior of CPRel was entirely opposite. All variants of CPRel, achieved the highest correlation on highest threshold value. For CPRel (Hybrid) and CPRel(LE) on R&G dataset , the lowest and highest correlation values were observed on the top two threshold values, though the difference among correlation values on all cutoffs was not very significant. For WordSimilarity-353 dataset, as indicated by Table. 5, the highest correlation value was achieved on 0.01 threshold for all variants of CPRel but for other thresholds, their correlation changed randomly. Overall, the performance of CPRel (LE) remained consistent on all three datasets with minor changes in correlation values. It was found that the performance of each dataset was different on different threshold values. This elucidated the fact the there is no unique threshold value which could be used as a discriminator for good or bad relatedness performance on all datasets.

In another experiment, performance of each approach with and without Context Filtering (CF) was compared. In case, when no CF was applied and all the words that matched to Wikipedia labels were considered, the behavior of each dataset was again different. For CPRel (NTF) there was a significant improvement in the correlation values (r_s) on all three datasets when CF was applied. On average, there was an increase of 15% in the correlation value on all three datasets, weighted by their sizes. In case of other two approaches, there is no significant improvement in correlation values with CF. It means that the CPRel (NTF) performs quite well with context filtering.

Table 6. Best performance of three variants of CPRel on three benchmark datasets with and without Context Filtering (CF)

Dataset	CPRel(hybrid)		CPRel(NTF)		CPRel(LE)	
	CF	No CF	CF	No CF	CF	No CF
M&C	0.83	0.83	0.83	0.72	0.81	0.81
R&G	0.79	0.78	0.79	0.62	0.79	0.78
WordSimilarity-353	0.66	0.64	0.69	0.54	0.64	0.62

It was also found that the relatedness computation not only depends on the context filtering but also on the nature of the dataset and type of weighting scheme. In comparison with other existing Wikipedia based approaches, shown in Table. 7, CPRel performed significantly better then ESA on M&C dataset. For other two datasets, it performed better then other Wikipedia based approaches but was still behind ESA. There are two reasons for this: First, to identify the context of each input word CPRel focus only on corresponding Wikipedia article contents whereas, ESA makes a good use of whole Wikipedia corpus to mine the context of each word. This is a limitation of CPRel and in future it is intended to modify this approach so that the context spread of each word in the whole corpus may effectively be used to improve relatedness computation. Second, CPRel approach works quite well on noun word pairs in particular, justifying best performance on M&C dataset (which is a noun subset of R&G dataset). The advantage of CPRel is that it does not require preprocessing of the whole Wikipedia corpus like ESA which is computationally quite expensive and laborious. It is proved that good relatedness scores can be achieved following a simple and computationally inexpensive approach. Another advantage of CPRel is that it can be effectively used for document relatedness also. As a future work, it is intended to test document relatedness based on the same approach. To top this, other features of Wikipedia such as hyperlink structure, category network and corpus statistics etc. could be considered to improve context profiles for getting better relatedness scores.

Table 7. Best performance comparison of CPRel with existing Wikipedia based approaches on three benchmark datasets

Dataset	WikiRelate	ESA	WLM	**CPRel(proposed)**
M&C	0.45	0.73	0.70	**0.83**
R&G	0.52	0.82	0.64	**0.79**
WordSim-353	0.49	0.75	0.69	**0.64**

5 Conclusions

This paper describes CPRel, a measure of semantic relatedness using Wikipedia. Semantic relatedness of words is computed by constructing context profile of each word based on Wikipedia article content and labels. The influence of different factors like types of weighting scheme, nature of dataset, nature of knowledge source and impact of cutoff threshold value on the performance of relatedness computation was analyzed. The impact of a cutoff threshold LP value on each weighting scheme was tested using various threshold values and it was found that context filtering was helpful in improving the relatedness scores in case of CPRel (NTF). When evaluated on three benchmark datasets of term relatedness, CPRel performed quite well in comparison with other Wikipedia based approaches.

References

1. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: AAAI. (2011)
2. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: Proceedings of Association for the Advancement of Artificial Intelligence(AAAI). (2006)
3. Barzilay, R., Elbadad, M.: Using lexical chains for text summarization (1997)
4. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. In: Readings in information retrieval. Morgan Kaufinann, USA (1997) 323–328
5. He, X., Ding, C.H.Q., Zha, H., Simon, H.D.: Automatic topic identification using webpage clustering. In: Proceedings of IEEE International Conference on Data Mining(ICDM). (2001) 195–202
6. Coursey, K., Mihalcea, R.: Topic identification using wikipedia graph centrality. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL-Short '09 (2009) 117–120
7. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. CIKM '08 (2008) 509–518
8. Medelyan, O., Witten, I.H., Milne, D.: Topic indexing with Wikipedia. In: Wikipedia and Artificial Intelligence: An Evolving Synergy. Papers from the 2008 AAAI Workshop, Menlo Park, CA, USA, Proceedings of AAAI (2008) 19–24
9. Agirre, E., Unibertsitatea, E.H., Rigau, G.: A proposal for word sense disambiguation using conceptual distance. In: Proceedinds of 1st international conference on recent advances in natural language processing (RANLP). (1990)
10. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico (2003) 241–257
11. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on information and knowledge management. CIKM '07, New York, NY, USA (2007) 233–242
12. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining. (2000)
13. Budanitsky, A., Hirst, G.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: Workshop on wordNet and other lexical resources, second meeting of the North American Chapter of the Association for Computational Linguistics. (2001)
14. Dagan, I., Lee, L., Pereira, F.C.N.: Similarity-based models of word cooccurrence probabilities. Journel of Machine Learning Research **34** (1999) 43–69
15. Lee, L.: Measures of distributional similarity. In: Proceedings of the ACL. (1999) 25–32
16. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. American Society for Information Science, JASIS **41** (1990) 391–407
17. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '99, New York, NY, USA (1999) 50–57

18. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
19. Sun, Q., Li, R., Luo, D., Wu, X.: Text segmentation with lda-based fisher kernel. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Stroudsburg, PA, USA (2008) 269–272
20. Yeh, E., Ramage, D., Manning, C.D., Agirre, E., Soroa, A.: Wikiwalk: random walks on wikipedia for semantic relatedness. In: 2009 Workshop on Graph-based Methods for Natural Language Processing. TextGraphs-4 (2009) 41–49
21. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. (2007) 1606–1611
22. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy. (2008) 25–30
23. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th international conference on World wide web. WWW '11, New York, NY, USA (2011) 337–346
24. Jabeen, S., Gao, X., Andreae, P.: Harnessing wikipedia semantics for computing contextual relatedness. In: PRICAI 2012: Trends in Artificial Intelligence. Volume 7458 of Lecture Notes in Computer Science. Springer (2012) 861–865
25. Halawi, G., Dror, G., Gabrilovich, E., Koren, Y.: Large-scale learning of word relatedness with constraints. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '12, New York, NY, USA (2012) 1406–1414
26. Liu, Y., McInnes, B.T., Pedersen, T., Melton-Meaux, G., Pakhomov, S.: Semantic relatedness study using second order co-occurrence vectors computed from biomedical corpora, umls and wordnet. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. IHI '12, New York, NY, USA (2012) 363–372
27. Navigli, R., Ponzetto, S.P.: Babelrelate! a joint multilingual approach to computing semantic relatedness. In: AAAI. (2012)
28. Milne, D.: An open-source toolkit for mining wikipedia. In: Proceeding of New Zealand Computer Science Research Student Conference. Volume 9. (2009)
29. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Comput. Linguist. **32** (2006) 13–47
30. Zesch, T., Gurevych, I.: Automatically creating datasets for measures of semantic relatedness. In: Proceedings of the Workshop on Linguistic Distances, ACL. (2006) 16–24

Evaluation and Comparison of multilingual fusion strategies for similarity-based Word Sense Disambiguation

Andon Tchechmedjiev, Jérôme Goulian, and Didier Schwab

Univ. Grenoble Alpes
Grenoble, France
{andon.tchechmedjiev,jerome.goulian,didier.schwab}@imag.fr
http://getalp.imag.fr/wsd

Abstract. In this article, we investigate the effects on the quality of the disambiguation of exploiting multilingual features with a similarity-based WSD system based on an Ant Colony Algorithm. We considered features from one, two, three or four languages in order to quantify the improvement brought by using features from additional languages. Using BabelNet as a multilingual resource, we considered three data fusion strategies: an early fusion strategy, and two late fusion strategies (majority vote and weighted majority vote). We found that the early fusion approach did not produce any significant improvements while voting strategies adding features from more languages led to an increase in the quality of the disambiguation of up to 2.84%. Furthermore, a simple majority vote led to better results than the weighted variant.

Key words: Similarity-based WSD, Multilingual WSD, Multilingual Features, BabelNet, Fusion Strategies

1 Introduction

Word Sense Disambiguation (WSD) is a core problem in Natural Language Processing (NLP), as it may improve many of its applications, such as multilingual information extraction, automatic summarization or machine translation. More specifically, the aim of WSD is to find the appropriate sense(s) of each ambiguous word in a text from a pre-defined sense inventory.

Focused mainly on the disambiguation of English language text, many WSD algorithms exist and can be classified according different typologies. Figure 1 presents various types of WSD methods according to two dimensions: the use of sense-annotated corpora (y axis) and the use of large lexical databases (x axis). The more an approach is located towards the top of the figure, the more it requires sense-annotated corpora; the more an approach is located towards the right of the figure, the more the quality and size of the lexical databases used is important. Furthermore, the more an approach is located in the upper right corner, the more it will lead to good disambiguation results. Building a

good quality large scale lexical resource and large annotated corpora for one specific language is obviously very costly and one can consider that such resources currently only exist for English.

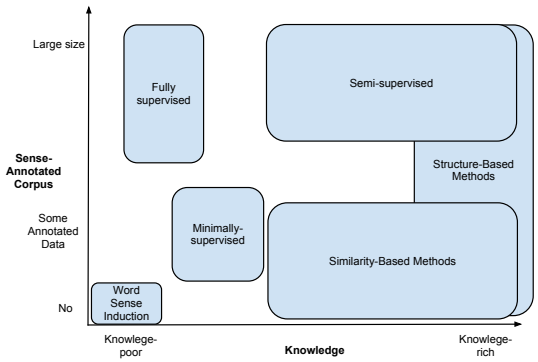


Fig. 1: One WSD approaches classification

In this article, we will focus on similarity based methods. These methods assign scores to word senses through semantic similarity (between word senses), and globally find the sense combinations maximising the score over a text. In other words, a local measure is used to assign a similarity score between two lexical objects (senses, words, constituencies) and the global algorithm is used to propagate the local measures at a higher level.

One of the first knowledge-based method was proposed by [1] and assesses semantic similarity as the number of words in common between sense definitions. The Lesk measure is still widely used today, however, in its original form, suffers from the small size of word-sense definitions. Methods have been proposed to partly solve the issue, either by exploiting the definitions of related senses from a structured lexical resource [2] or by considering the surrounding text in the computation of the overlap [3].

The objective of this article is to evaluate if exploiting multilingual features leads to improved results. We successively experiment adding one, two or three language to our WSD system based on a global ant colony algorithm. In the next sections, we first present a general account of data fusion strategies and then describe the semantic similarity measure and the ant colony algorithm that we use. Subsequently we will describe the method we use to generate both monolingual and combined multilingual dictionaries for our extended Lesk measure (early fusion), followed by a description of the different voting strategies we considered, our experimental protocol and the results.

2 Contribution of multilingual informations for WSD

Using multilingual information for Word Sense Disambiguation can be achieved in two ways: either combining multilingual information before the execution of the disambiguation algorithm (early fusion) or combining multilingual information after the execution of the disambiguation algorithm (late fusion).

The concepts of early and late fusion come from the field of classification and were introduced to deal with the combination of multiple sources of information. The objective of information fusion is thus to improve the performance and robustness of classification systems relying on multi-modal or multiple-source mono-modal input([4], Chapter 4.3).

2.1 Early and late fusion

One of the first works accounting of a combination of multilingual features using an early fusion approach has been proposed by [5], where a unigram word frequency vectorial representation is used. A monolingual vector for each word sense is first created from an english definition. An MT system (Google translate) is then used to obtain features in French, German and Spanish that are subsequently added to the vectors. Each vector is then weighted with a normal distribution for each of the languages. For the evaluation, combinations of English with one, two or three other languages are considered. The model leads to a reduction in the number of classification errors between 10.58% and 25.96%. Moreover, the more languages are added, the better are the results obtained. More recently, [6] have proposed a similar approach, albeit not with a vectorial model, but derived from a simplified lesk algorithm [3] that selects the word senses that maximise the overlap with the words of the surrounding context rather than between combinations of word-senses. For each word sense, the context in English is translated using an MT system in order to obtain multilingual representations. The purpose is to compare the classification error rate reduction between using only one language (English, and through MT German, French, Spanish) to that of using a concatenated context of all four languages. The evaluation was made on the Semeval 2007 lexical sample task and reports a 26% maximal error rate reduction with four combined languages. Hasan et al. [7] have undertaken to evaluate the influence of adding features from multiple languages, but at the level of semantic relatedness measures using either a MT system or manual translation to generate features. They reach the conclusion that additional information can be acquired from multilingual features, thus significantly improving the correlation with human judgement. However they also find that the improvement is all the more significant for languages with scarce lexical resources (37% improvement for Romanian for example against 8% for English).

Late fusion approaches have mostly been used to combine monolingual features through classifier voting strategies. [8] evaluate the combination of different systems through several late fusion strategies on Senseval-1 English as well as Senseval-2 with 4 languages individually. They report improvements of up to

4% over the best published score for the respective campaigns. [9] have similarly evaluated a broad range of WSD algorithms on the Web-Harvested Corpus Annotated with GermaNet Senses, as well as combinations thereof using several voting strategies (majority vote, weighted majority vote, Borda count), which led to an improvement of 7.23% compared to the best performing standalone algorithm (a variant of Lesk). [10] evaluate two WSD systems, one that uses monolingual features only and one that exploits multilingual information extracted from parallel corpora; their combination through a voting strategy leads to an improvement of 2.24%, which constitutes an indication that multilingual features can indeed be combined beneficially through a vote strategy.

2.2 Motivations: Multilingual Word Sense Disambiguation

The results mentioned in the previous section clearly show that there is a lot of potential in using multilingual informations to improve WSD performance. Our work group currently focuses on multilingual Word Sense Disambiguation. The aim of this task is to find the appropriate sense(s) of each word of a text whatever its language (more reasonably for a certain number of languages). The evaluation requires a reference corpus in various languages. Such an evaluation will be carried out in the SemEval 2013 evaluation campaign (task 12: *Multilingual Word Sense Disambiguation*, march 2013). At the time of the writing,neither the corpus nor the gold standard of the campaign were available. Instead, we used the English all-words task 7 corpus [11] of the SemEval 2007 campaign, where organizers provided a coarse-grained pre-definied sense inventory from WordNet. Considering coarse grained senses distinction means that close senses are counted as equivalent (e.g. snow/precipitation and snow/cover).

3 The LIG-GETALP WSD System

The LIG-GETALP system is a similarity based system. At the local level (between two senses), we use a Lesk similarity measure and at the global level we use our own ant colony algorithm [12–14].

3.1 Local similarity measure: an adaptation of Extended Lesk

Our local algorithm is a variant of the Lesk similarity measure [1], where the score given to a sense pair is the number of words in common between the two definitions, without taking into account either the word order or any syntactic or morphological information.

In this work, we use the definitions from BabelNet, a large scale multilingual lexical resource built from WordNet and Wikipedia (see section 4).

3.2 Global algorithm: Ant Colony Algorithm

Ant colony algorithms (ACA) are inspired from nature through observations of ant social behaviour. Indeed, these insects have the ability to collectively find

the shortest path between their nest and a source of food (energy). It has been demonstrated that cooperation inside an ant colony is self-organised and allow the colony to solve complex problems. The environment is usually represented by a graph, in which virtual ants exploit pheromone trails deposited by others, or pseudo-randomly explore the graph. ACAs are a good alternative for the resolution of optimization problems that can be encoded as graphs and allow for a fast and efficient exploration on par with other search heuristics. The main advantage of ACAs lies in their high adaptivity to dynamically changing environments. Readers can refer to [15] or [16] for a state of the art.

In this article we use a simple hierarchical graph (text, sentence, word) that matches the structure of the text and that exploits no external linguistic information. In this graph we distinguish two types of nodes: nests and plain nodes. Following [13], each possible word sense is associated to a nest. Nests produce ants that move in the graph in order to find energy and bring it back to their mother nest: the more energy is brought back by ants, the more ants can be produced by the nest in turn. Ants carry an odour (vector) that contains the words of the definition of the sense of its mother nest. From the point of view of an ant, a node can be: (1) *its mother nest*, where it was born; (2) *an enemy nest* that corresponds to another sense of the same word; (3) *a potential friend nest*: any other nest; (4) a *plain node*: any node that is not a nest. Furthermore, to each plain node is also associated an odour vector of a fixed length that is initially empty.

Ant movement is function of the scores given by the local algorithm, of the presence of energy, of the passage of other ants (when passing on an edge ants leave a pheromone trail that evaporates over time) and of the nodes' odour vectors (ants deposit a part of their odour on the nodes they go through). When an ant arrives onto the nest of another word (that corresponds to a sense thereof), it can either continue its exploration or, depending on the score between this nest and its mother nest, decide to build a bridge between them and to follow it home. Bridges behave like normal edges except that if at any given time the concentration of pheromone reaches 0, the bridge collapses. Depending on the lexical information present and the structure of the graph, ants will favour following bridges between more closely related senses. Thus, the more closely related the senses of the nests are, the more bridges between them will contribute to their mutual reinforcement and to the sharing of resources between them (thus forming *meta-nests*); while the bridges between more distant senses will tend to fade away. We are thus able to build interpretative paths (possible interpretations of the text) through emergent behaviour and to suppress the need to use a complete graph that includes all the links between the senses from the start (as is usually the case with classical graph-based optimisation approaches).

4 BabelNet, a large scale multilingual resource

BabelNet [17] is a large scale multilingual lexical resource built from the automatic mapping between WordNet synsets and Wikipedia pages. BabelNet is

based on the concept of a *Babel Synset* that contains a WordNet synset and a set (or sets) of related Wikipedia pages. The latter include Wikipedia pages mapped to WordNet synsets (through a disambiguation algorithm), pages related to the mapped pages (inter-page hyper-links), as well as corresponding multilingual pages obtained through Wikipedia's inter-language links and their redirections towards pages on the Wikipedia in the respective languages. For concepts that are not covered in all languages supported by BabelNet (English, French, German, Italian, Spanish and Catalan), a set of related sentences are extracted from the text in linked Wikipedia pages and from Semcor instances [18] and are then translated in the missing languages through a machine translation system , thus guaranteeing a homogeneously high coverage. Figure 2 is an illustration of BabelNet's structure. Additionally, BabelNet provides glosses (definitions) associated to each Babel synset in all the supported languages, which is invaluably useful for Lesk-based approaches.

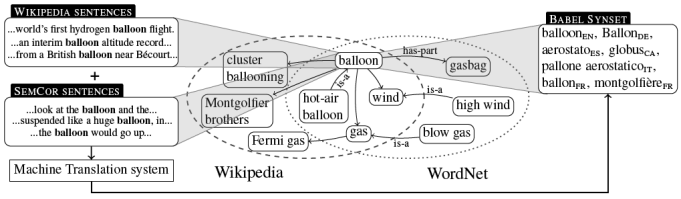


Fig. 2: An overview of BabelNet (figure 2 from [17])

5 Data fusion strategies

As mentioned in previous sections, there are two main data fusion strategies: either combining informations before the execution of the disambiguation algorithm (early fusion) or combining informations after the execution of the disambiguation algorithm (late fusion). In this article, early fusion is achieved by concatenating linked definitions in our lexical resource (BabelNet) and late fusion is achieved by combining the results of the disambiguation obtained for each language. Figure 3 provides illustrations of both strategies for our WSD similarity-based approach.

5.1 Early fusion, *a priori* generation of multilingual dictionaries

We first generate monolingual dictionaries for the four languages chosen for evaluation: English, French, German and Spanish, from which we obtain monolingual results from the execution of our disambiguation algorithm. These results will be used as baselines to evaluate the potential improvement brought by adding more

(a) Early fusion (b) Late fusion

Fig. 3: Illustrations of the fusion strategies for Knowledge-based WSD

languages. For the early fusion, we generated dictionaries for all the combinations of 1, 2, 3 and 4 languages (a total of 14 dictionaries) that were subsequently used for the experiments.

5.2 Late fusion voting strategies

As explained earlier, late fusion corresponds to a combination of the output of different classifiers. In the case of results generated by WSD systems for the Semeval All-words task 7, the output of the algorithm is sense assignments for the whole corpus, which exactly corresponds to the output of a classifier over a set of instances. However, the format of the results of the campaign only requires to provide sense labels for each word, which means we have no information on the confidence in the assignments or any form of posterior probability distribution. As such, only a simple majority voting mechanism is applicable [8]. Of course in the case of our ant colony algorithm, each node corresponding to a word sense has an associated energy level that we use to select the final answer. While it does not constitute a posterior probability distribution as such, it offers likelihood values that can be used as weights for a weighted majority vote.

So as to be able to make a fair comparison with other systems, we considered both a simple majority vote and a weighted majority vote.

Specifically in the case of the fusion of multilingual resources, the vote is performed on result files generated by the execution of the algorithm using each of the corresponding monolingual dictionaries as input. For example, in order to apply a late fusion strategy to combine French and English, one needs to make several executions with the dictionaries in both languages. Subsequently the voting strategy can be applied on the result files (in equal number) of French and English executions, to obtain a single result file.

In our experiments, we used 12 input files in total per vote, distributed equally among the languages. Of course this means that we need 12 times more executions to apply a late vote strategy, which constitutes a significant amount of time. In order to mitigate this problem, over 120 executions, we applied the vote on 12 successive answers by increments of 1 modulo 120 so as to obtain 120 vote results without requiring more executions. Even though the samples are somewhat correlated, we found no evidence of impact on the quality of the results obtained compared to always using separate executions.

For the majority vote strategy, for each word of the corpus, we build an histogram of the cumulative distribution of selected senses among the result files. Subsequently, we select for each word the sense with the relative majority (maximum frequency).

For the weighted vote, instead of giving only a single vote for one sense selection, several votes (the weight values) are given for each sense selection. In the case of our algorithm, we had a different energy value for each sense in each result file that we used as a weight. Like for the unweighted majority vote, for each word, the sense with the maximum score (sum of energy values) is selected.

The next section describes more precisely the experiments and their outcome.

6 Experimental evaluation and results

6.1 Experimental Protocol

The objective of our experiments is to determine if multilingual features do indeed improve the quality of disambiguation, and, if so, whether or not the number of languages have an influence on the degree of improvement. Furthermore, we want to know, which, among early fusion and two late fusion techniques for the combination of features, yields the best and most salient results.

Since our algorithm is stochastic in nature, we need to have a representation of the distribution of solutions as accurate as possible in order to make a statistically significant analysis. Thus, we used 120 samples for each of the groups of one, two, three and four languages. In order to avoid any bias we systematically considered all possible language combinations for 1, 2 or 3 languages and the whole language group (the 4 languages together). Depending on the number of combinations, we made for each combination a number of runs so as to obtain 120 executions in total for the whole language group. For example for the 1 language group, we considered 30 executions of English, French, German and Spanish and grouped them together.

In order to study the distribution of the scores depending on each language group, we used a one-way ANOVA variance analysis, coupled with a Tukey's HSD post-hoc pairwise analysis. The one-way ANOVA analysis is a generalization to several groups of the unpaired t-test that compares the means of only two samples. This method offers a way to test the hypothesis of equality of the mean scores at different levels, e.g between the different groups of languages. If the difference is statistically significant (with a p-value $p < 0.05$), one needs to determine which are the language pairs whose scores are significantly different; Tukey's method is a way to make all pairwise comparisons of mean scores using studentized range distribution to prevent the accumulation of type I errors (incorrect rejection of a true null hypothesis).

6.2 Results

Figure 4 and Tables 1 and 2 present the average scores and standard deviations for: early fusion, the majority vote and weighted majority vote respectively.

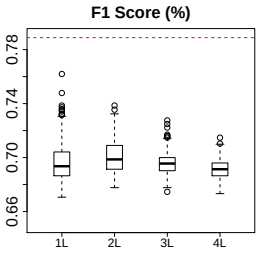
Similarly, Figures 4, 5a and 5b, present box plots of the results; the dashed line is the WordNet first sense baseline. The parenthesis with a † lists the language groups with significant differences in the average score (Tukey's HSD test).

In order to check the normality assumption for ANOVA, we computed the correlation between the theoretical (normal distribution) and the empirical quantiles. For all language combinations there always was a correlation above 0.99. Furthermore we used Levene's variance homogeneity test and found a minimum significance level of 10^{-6} between all algorithms and metrics.

Before analysing the results, it is important to note that the purpose of this experiment is not to obtain results better than the state of the art, but rather to see the general effect of adding multilingual features without any consideration of a particular language. Indeed, for reference, in a purely monolingual setting, we obtain on average above 77% using a dictionary generated from WordNet only and above 79% using a majority vote strategy. We have noted a decreased score when using only BabelNet compared to using WordNet on English for our WSD algorithm with the current parameters, however for languages other than English, the results remain very interesting. Let us first look at Figure 4 that present the results for the early fusion of multilingual features.

Algorithm	F_1 (%)	σ_{F_1}
F.S. Baseline	78.89	N/A
1 Lang.	69.92 †$(4L)$	0.0181
2 Lang.	70.07 †$(3L, 4L)$	0.0132
3 Lang.	69.59 †$(2L, 4L)$	0.0096
4 Lang.	69.12 †$(1L, 2L, 3L)$	0.0076

(a) F_1 scores, († $\leftrightarrow p < 0.05$) for each language group for early fusion



(b) Boxplot of F_1 scores for early fusion *vs.* WordNet first sense baseline (dashes)

Fig. 4: Results for early fusion

For one language, the average score is quite low, below 70%. However, we can see that the results over English are notably higher than the others (indicated by the many outliers at the top of the distribution). When adding more languages, the general tendency is that the score decreases but also the standard deviation. We can see that the more languages are added, the less outliers there are on the top of the distribution. Furthermore, there are no significant improvements when going from one to two languages. Adding more languages has the effect of causing a slight but significant decrease of the average towards the bottom of the distributions (they roughly share the same minimal value).

We hypothesize that because we only compute our local score as a non-normalised bag of words overlaps and because the number of related *Babel*

synsets can vary, we find ourselves confronted with definitions of very different lengths. Thus, given that longer definitions are privileged, a strong bias is added to the system. Furthermore, given the automatic construction of BabelNet, we think that there may be a higher degree of noise compared to just WordNet, which may very well cause a shift in the parameter space and thus lead to bad scores for parameters that worked much better when only WordNet was used.

For the results of the majority vote strategy, let us look at table 1 and figure 5a. Compared to the one language case with no vote (69.92%), the vote strategy leads to a leap in the score of 4.14%, which is in accordance with the improvements observed when using voting strategies in a monolingual setting [8, 9]. In terms of the improvement when adding more languages, we can see a steady and significant increase in the average score of +1.73% for two languages and +2.32% for three, while a fourth language did not lead to any significant improvements compared to three languages. Furthermore, adding more languages leads to a more compact distribution: between one language and four languages the standard deviation is divided by six.

For the weighted majority vote (table 2 and figure 5b), we can make the same observations about the effect of adding more languages, except that the scores are notably lower than with a simple majority vote. Indeed, for one language, the difference in the average scores between the majority vote and the weighted majority vote is -1.63%, for two languages -1.86%, for three languages -1.45% and for four languages -1.27%. It appears that the majority vote leads to better results in the WordNet monolingual setting. Given that the majority vote is reported as well to show better results by [9], it may be beneficial to consider other simple weighting schemes, for example different weights per part of speech.

Algorithm	F_1 (%)	σ_{F_1}
F.S. Baseline	78.59	N/A
1 Lang.	74.06 †$(2L, 3L, 4L)$	0.020
2 Lang.	75.79 †$(1L, 3L, 4L)$	0.0172
3 Lang.	76.38 †$(1L, 2L)$	0.0112
4 Lang.	76.90 †$(1L, 2L)$	0.0033

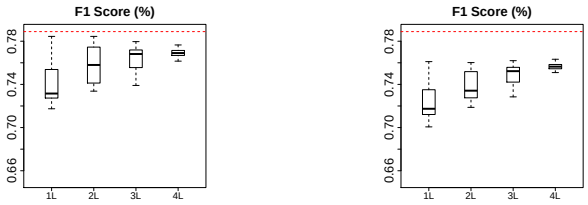
Table 1: Comparison of the F_1 scores, ($\dagger \leftrightarrow p < 0.05$) for each language group for majority vote

Algorithm	F_1 (%)	σ_{F_1}
F.S. Baseline	78.59	N/A
1 Lang.	72.43 †$(2L, 3L, 4L)$	0.0172
2 Lang.	73.94 †$(1L, 3L, 4L)$	0.0126
3 Lang.	74.93 †$(1L, 2L, 4L)$	0.0083
4 Lang.	75.63 †$(1L, 2L, 3L)$	0.0028

Table 2: Comparison of the F_1 scores, ($\dagger \leftrightarrow p < 0.01$) for each language group for weighted majority vote

7 Conclusion and perspectives

In this article, we evaluated on the all-words WSD task of Semeval 2007 the effects on disambiguation quality of exploiting multilingual features with our WSD system based on an Ant Colony Algorithm. Multilingual informations are provided by BabelNet, a lexical multilingual resource available since 2012, and built automatically in aligning Wikipedia and WordNet. Three data fusion strategies



(a) Majority Vote (b) Weighted Majority Vote

Fig. 5: Box plots of the F_1 scores for majority vote (a) and weighted majority vote (b) compared to the WN FS baseline (dashed line)

have been used: an early fusion strategy, and two vote-base strategies (majority vote and weighted majority vote). We have found that with the early fusion strategy adding multilingual features did not lead to any significant improvements. Integrating information from BabelNet directly lead to poorer scores than with just WordNet. We believe the main cause is that BabelNet is slightly noisier than WordNet. Our ant colony algorithm requiring a minimal level of parameterization, we believe that the parameters that offered very good results with WordNet were far from optimal when using a lexical resource where a significant number of definitions are built with Machine Translation Systems. However, when using late fusion strategies (voting strategies), there was a clear improvement when adding multilingual features from more and more languages. Possible approaches to improve the results we obtained with early fusion would be to consider different ways of concatenating the glosses, for instance imposing a global maximum length for concatenated definitions. Another approach would be to consider the fusion at the feature level, and to use several monolingual dictionaries at a time in our ant colony algorithm by having different types of ant that use features from a different language and to combine the features in the environment through emergence.

Acknowledgments

The work presented in this paper was conducted in the Videosense project, funded by the French National Research Agency (ANR) under its CONTINT 2009 programme (grant ANR-09-CORD-026).

References

1. Lesk, M.: Automatic sense disambiguation using mrd: how to tell a pine cone from an ice cream cone. In: Proceedings of SIGDOC '86, New York, NY, USA, ACM (1986) 24–26
2. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: CICLing 2002, Mexico City (2002)

3. Vasilescu, F., Langlais, P., Lapalme, G.: Evaluating variants of the lesk approach for disambiguating words. In: Proceedings of LREC 2004, Lisbon, Portugal (2004)
4. Bezdek, J., Keller, J., Krishnapuram, R., Pal, N.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers (1999)
5. Banea, C., Mihalcea, R.: Word sense disambiguation with multilingual features. In: Proceedings of ICCS'2011, Oxford, United Kingdom, ACL (2011) 25–34
6. Fernandez-Ordonez, E., Mihalcea, R., Hassan, S.: Unsupervised word sense disambiguation with multilingual representations. In: Proceedings of LREC'2012, Istanbul, Turkey (2012)
7. Hassan, S., Banea, C., Mihalcea, R.: Measuring semantic relatedness using multilingual representations. In: Proceedings of SemEval 2012, Montréal, Canada, ACL (2012) 20–29
8. Florian, R., Yarowsky, D.: Modelling consensus: classifier combination for word sense disambiguation. In: Proceedings of EMNLP '2002, University of Pennsylvania, Philadelphia, ACL (2002) 25–32
9. Henrich, V., Hinrichs, E.: A comparative evaluation of word sense disambiguation algorithms for german. In: Proceedings of LREC'2012, Istanbul, Turkey, ELRA (2012)
10. Guo, W., Diab, M.: Combining orthogonal monolingual and multilingual sources of evidence for all words wsd. In: Proceedings of ACL'2010, Uppsala, Sweden (2010) 1542–1551
11. Navigli, R., Litkowski, K.C., Hargraves, O.: Semeval-2007 task 07: Coarse-grained english all-words task. In: SemEval-2007, Prague, Czech Republic (2007) 30–35
12. Schwab, D., Goulian, J., Guillaume, N.: Désambiguïsation lexicale par propagation de mesures semantiques locales par algorithmes a colonies de fourmis. In: TALN, Montpellier (France) (2011)
13. Schwab, D., Goulian, J., Tchechmedjiev, A., Blanchon, H.: Ant colony algorithm for the unsupervised word sense disambiguation of texts: Comparison and evaluation. In: Proceedings of COLING'2012, Mumbai (India) (2012) To be published.
14. Tchechmedjiev, A., Goulian, J., Schwab, D., Sérasset, G.: Parameter estimation under uncertainty with simulated annealing applied to an ant colony based probabilistic wsd algorithm. In: Proceedings of the First International Workshop on Optimization Techniques for Human Language Technology, Mumbai, India, The COLING 2012 Organizing Committee (2012) 109–124
15. Dorigo, Stützle: Ant Colony Optimization. MIT-Press (2004)
16. Monmarché, N.: Artificial Ants. Iste Series. John Wiley & Sons (2010)
17. Navigli, R., Pozetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence (2012) http://dx.doi.org/10.1016/j.artint.2012.07.004.
18. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: Proceedings of the workshop on Human Language Technology. HLT '93, Stroudsburg, PA, USA, Association for Computational Linguistics (1993) 303–308

A Novel Approach for Detecting Arabic Persons' Names using Limited Resources

Omnia Zayed, Samhaa El-Beltagy, and Osama Haggag

Center of Informatics Science, Nile University, Giza, Egypt

{omnia.zayed, samhaaelbeltagy, osama.haggag}@gmail.com

Abstract. Named entity recognition is an involved task and is one that usually requires the usage of numerous resources. Recognizing Arabic entities is an even more difficult task due to the inherent ambiguity of the Arabic language. Previous approaches that have tackled the problem of Arabic named entity recognition have used Arabic parsers and taggers combined with a huge set of gazetteers and sometimes large training sets. However, the recent surge in the usage of social media, where colloquial Arabic, rather than modern standard Arabic is used, invalidates these approaches because existing parsers fail to parse colloquial Arabic at an acceptable level of precision. To address such limitations, this paper presents an approach for recognizing Arabic persons' names without utilizing any Arabic parsers or taggers. The approach uses only a limited set of publicly available dictionaries. The followed approach integrates dictionaries with a statistical model based on association rules for extracting patterns that indicate the occurrence of persons' names. Through experimentation on a benchmark dataset, we show that the performance of the presented technique is comparable to the state of the art machine learning approach.

Keywords: Arabic Named Entity Recognition, Association Rules, Colloquial Arabic, Modern Standard Arabic.

1 Introduction

The importance of named entity recognition (NER) is increasing progressively due to the necessity of a better understanding of human communication. A lot of applications in the field of Natural Language processing make use of NER as extensively listed in [12]; examples of those applications include Machine Translation, Text Clustering and Summarization, Information Retrieval and Question Answering systems.

Approaches for recognizing named entities from text, fall under three categories. The first approach known as "rule based NER" combines grammar, in the form of handcrafted rules, with gazetteers to extract named entities. The second, is "machine learning based NER" which utilizes large datasets and features extracted from these, to train a classifier to recognize a named entity. Hence this approach converts the named recognition task into a classification task. Machine learning algorithms could be further categorized as either supervised or unsupervised. The third and final ap-

proach is "hybrid NER" which combines both of the aforementioned approaches [18, 25]. A comparison between rule based approaches and machine learning approaches is presented in [18] in terms of the used domain. The reason behind the difficulty of modifying rule based approach for new domains has been related to the use of a lot of resources such as gazetteers, besides the need of complicated linguistic analysis to detect the named entities. On the other hand, machine learning approaches need a training dataset which is tagged in a certain manner to recognize new entities from new testing dataset of the same domain. Besides a precise selection of features is required [1, 18, 25].

Building a system to extract Arabic named entities is a difficult task. Being a Semitic language, the Arabic language is well known for its complex morphology. In addition, Arabic has a unified orthographic case; it does not have capital letters. Conversely, in the English language which allows mixed letter cases; some named entities can be distinguished because they are capitalized. These include persons' names, locations and organization. Moreover, Arabic is notable for its inherent ambiguity in which one word could imply variety of meanings [17, 25]. The fact that many names are derived from adjectives complicates the task of recognizing persons' names even further. The distinctive challenges of Arabic language including ambiguity and complexity are explained in detail in [1].

While most existing Arabic texts are written in formal Modern Standard Arabic Text (MSA), the volume of informal colloquial Arabic text is increasing progressively with the wide spread use of social media examples of which are Facebook, Google Moderator and Twitter. Previous approaches that have tackled the problem of Arabic named entity recognition frequently depend on Arabic parsers and taggers combined with a huge set of gazetteers and sometimes large training sets to achieve their task. However, the task of named entities extraction from colloquial Arabic text invalidates these approaches as existing parsers fail to parse colloquial Arabic at an acceptable level of precision. This is due to sentence irregularity, incompleteness and the varied word order of colloquial Arabic. Colloquial Arabic also has no standard rules or grammatical constructions because it maps to a spoken language [25].

To address such challenges, this paper introduces an approach to recognize Arabic persons' names without utilizing any Arabic parsers or taggers. Moreover the followed approach tries to overcome the ambiguity problem of persons' names by organizing publicly available dictionaries of person names into clusters as will be detailed. Since the presented approach makes use of a limited set of dictionaries, integrated with a statistical model based on association rules, the model can easily generalize to different domains in our future work.

The rest of this paper is organized as follows: section 2 describes the proposed approach in detail. In section 3, system evaluation on a benchmark dataset is discussed. Section 4 reviews an overview of the literature on NER systems in Arabic language. Finally conclusion is presented in section 5.

2 The Proposed Approach

In this work, a rule based approach combined with a statistical model, is adopted in a novel way to identify and extract person names from Arabic text. Our approach tries to overcome two of the major shortcomings of using rule based techniques which are the difficulty of modifying a rule based approach for new domains and the necessity of using huge set of gazetteers. The proposed approach builds a statistical model for automatically extracting patterns which indicates persons' names occurrences, scored using association rules. Moreover, the ambiguity problem of persons' names is overcome using a clustering technique. Our approach consists of two phases, as shown in Figure 1. In the first phase, "The building of resources phase", person names are collected and clustered, and "name indicating" patterns are extracted. In the second phase, "Extraction of persons' names phase", name patterns and clusters are used to extract persons' names from input text. Both of these phases are described in depth, in the following subsections.

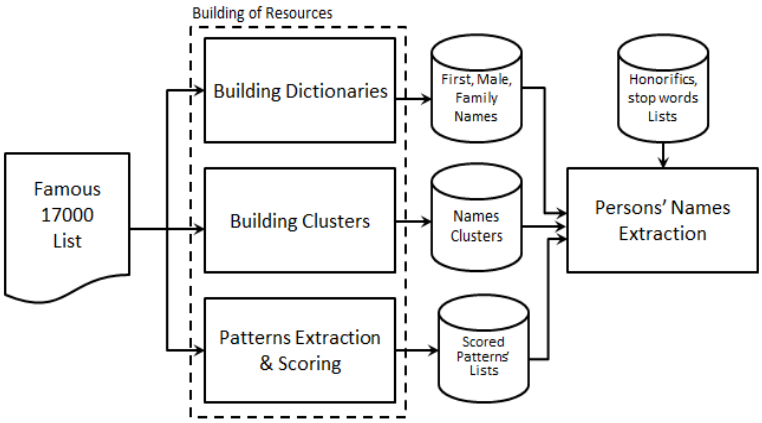


Fig.1. System Architecture.

2.1 The Building of Resources Phase

In this phase the resources on which the system depends are prepared. This phase is divided into 4stages.In the first stage, persons' names are collected from public resources. In the second stage, dictionaries of first, middle/male and family persons' names are built from collected resources. In the third stage, names are grouped together into clusters to avoid the Arabic persons' names ambiguity problem as will be detailed later. In the fourth and final stage, a corpus is used to build and score patterns which indicate the occurrence of a person name. Scoring of the patterns is done using association rules.

Persons' Names Collection. Wikipedia, with its huge collection of names under the people category, offers an excellent resource for building a name database. Kooora, which is an Arabic website for sports, also provides a large list of names. So,

in this stage, Wikipedia[1] and Kooora[2] websites were used to collect a list of about 17,000 persons' names. Since the aim of this work is not just to recognize names of famous people, but instead to identify the name of any person even if it does not appear in the collected list, the collection is further processed and refined in order to achieve this goal in the following step.

Building of Dictionaries. In this stage, the list of names collected in the previous step (we call this list the "famous_17000_list".) is processed in such a way so as to separate first names from family names. The processing step includes handling the different variations of Arabic persons' names. As described in [25], Arabic name could have affixes such as prefixes or embedded nouns. A word preceded or followed by those affixes must not be split on white spaces, instead the word and its affix should be considered as a single entity. For example, the male name عبدالعزيز (abdulaziz) should not be split as عبد (abd) as first name and العزيز (alaziz) as family name, instead it should be treated as single entity عبدالعزيز (abdulaziz) and considered as a first name. Table 1 lists the different variations of Arabic persons' names with examples [25].

Table 1. Different variations of writing Arabic persons' names.

Case	Example
Simple case (no affixes)	احمد محمود Ahmad Mahmoud
Prefix case {عبدAbd, ابو Abou, بن Bin, ال Al, …etc}	**عبد** العزيز **ال** سعود Abdulaziz Al Saud
Double prefix case {ابوعبدAbou Abd,بن عبد Bin Abd, … etc}	سلطان**بن عبد** العزيز **ال** سعود Sultan bin Abdulaziz Al Saud
Embedded noun case {الدين El-Deen,الله Allah, …etc}	هيردي نور **الدين** Herdi Noor Al-Din
Complex name (prefix + embedded noun)	تقي **الدين** محمد **بن** معروف الشامي Taqi al-Din Muhammad Ibn Ma'ruf al-Shami

Building of Name Clusters. Once names dictionaries are built, they can be used to identify previously unseen names by stating that a full name is composed of a first name followed by other male names and/or a family name. However, the problem is not that simplistic. One of the problems of rule based NER systems is that straight forward matching of persons' names using dictionaries, can often result in mistakes. For example, a phrase such as في خطاب بوش (In Bush's speech) a full name could be mistakenly extracted as خطاب بوش (Khatab Bush) even though it is highly unlikely that an Arabic person's name such as خطاب (Khatab) will appear besides an American person's name such as بوش (Bush). Arabic text often contains not only Arabic names, but names from almost any country transliterated to Arabic.

[1] http://ar.wikipedia.org/wiki/تصنيف:تراجم

[2] http://www.kooora.com/default.aspx?showplayers=true

The "famous_17000_list" thus contains Arabic, English, French, Hindi, and Asian persons' names, written in Arabic language. In our approach, clustering is used to separate these names. Clustering this list is an important step to determine acceptable name combinations. To carry out clustering, we have used the Louvain [13] graph clustering technique from within Gephi [6] which is an open source software for exploring and manipulating networks.

As a pre-processing step, the 17,000 persons' names list processed to build a dictionary in which each first name is a key item whose corresponding value is a list of the other family names it had, accounting for redundancies. The variations of writing Arabic persons' names mentioned in the previous sub-section is considered. This dictionary is converted to a graph, such that each first name and family name form separate nodes. Edges are then established between each first name and its corresponding family names. The resulting graph consisted of 15782 nodes, and 16481 undirected edges.

Then, the Louvain method was applied to the graph for finding communities within the network. The community in this context is a cluster of names that are related. A resolution parameter of 3.5waschosen, allowing larger communities to be found. The outcome was a set 2116 clusters, where each name is given a modularity class number denoting which community (cluster) it belongs to.

Figure 2 shows a snapshot of the resulting clusters. It was observed from visualizing the data that most of the culturally similar names were grouped together, for example it can be noted that most of the names common in the Arabic-speaking regions were grouped together. The same applies to English and French names and to other names that are kind of unique to their region. It was also observed that small lone clusters are those that contain rare names that do not have connections to the other names.

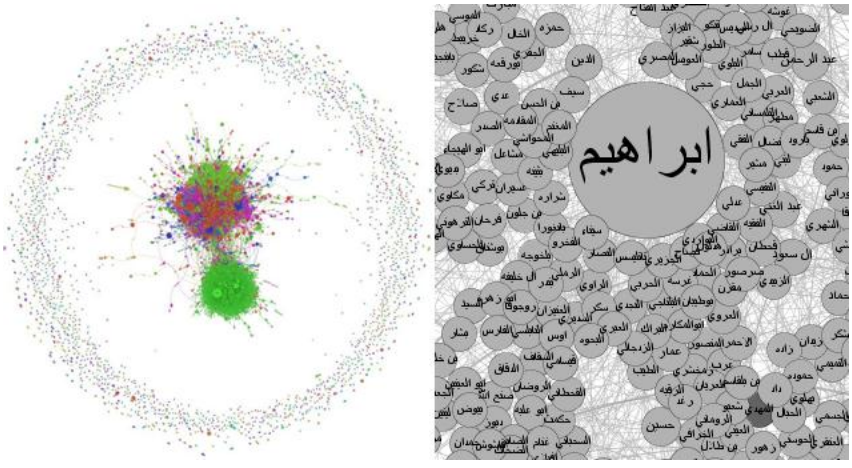


Fig.2. Visualization of generated clusters, to the left are all generated, lone clusters can be seen on the border and the two largest clusters are those of Arabic names (below) and Western names (above). To the right is a closer view of a subset of the Arabic names cluster.

Extracting Scored Patterns. In this stage, a statistical model is built to automatically learn patterns which indicate the occurrence of a person name. This stage is divided into 4 steps, as shown in Figure 3.

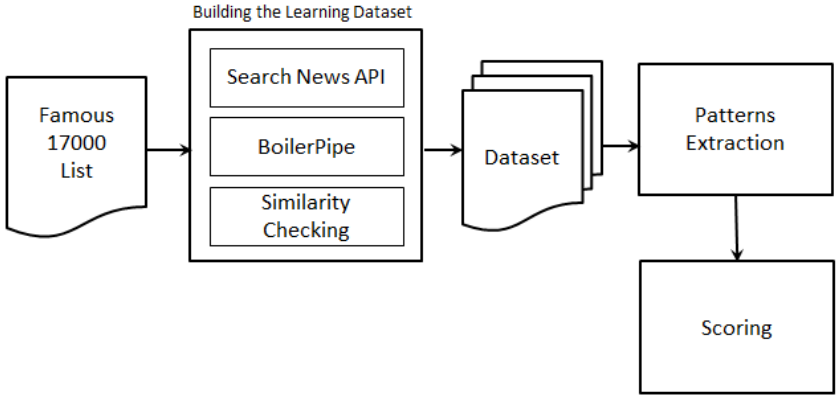


Fig.3. Building lists of scored patterns stage

Initially each name in the"famous_17000_list" is used as a query to search news articles to build learning dataset from the same domain that we are targeting to extract persons' names from. Akhbarak[3] API and Google Custom Search API[4]were used to search and retrieve news stories.

Around 200 news article links are crawled for each person name in the "famous_17000_list. After downloading the pages associated with these links BoilerPipe[5] is used to extract the main news article. Often news stories are repeated in many sources, so very similar stories are detected and removed.

Following this step, unigram patterns around each name are extracted. Three lists are formed. A complete pattern list keeps set of complete patterns around the name with their count. A complete pattern consists of <$word_1$><name><$word_2$>. The <name> part just indicates that a name has occurred between words: $word_1$ and $word_2$. Two type of unigram pattern lists are kept: a "before" list keeps the patterns that appear before a name with their counts (example: اكد (confirmed)) and an "after" list stores patterns that occur after a name with their count (example:ان (that)).

Finally the support measure employed by association rules [5] is used to score each pattern in the three lists. Support is calculated as the ratio of the count of a pattern followed by a name over the total count of all patterns followed by a name. For example the support rule used to score a unigram pattern before a name is computed by the following equation.

$$support_{patternbeforename} = \frac{Countthispatternfollowedbyaname}{Countoftotalsumofallpatternsfollowedbyaname} \quad (1)$$

[3] http://www.akhbarak.net/
[4] https://developers.google.com/custom-search/v1/overview
[5] http://code.google.com/p/boilerpipe/

The newly created three lists of scored patterns are saved descendingly according to the value of the score.

2.2 Extraction of Persons' Names Phase

The persons' names extraction process is dependent on the previous pre-prepared resources which are the dictionaries of first, and family names, the name clusters, a list of honorifics, a list of stop words and the patterns lists. Rules are implemented to extract persons' names from the unseen dataset of the same targeted domain. The benchmark dataset, ANERcorp[6]is used to evaluate the proposed system. The system assumes that any full name consists of a first name followed by one or more male names followed by zero or one family name. The generated name clusters are used to ensure that all candidate portions of a name fall in the same cluster to avoid matching mistakes. One of the rules used in the extraction phase is as follows:

```
For each word w_i in the target text:
  If w_i in patterns_before_list
    If w_i+1 in honorific_list
    Check for names from w_i+2 in the same cluster;
  Stop when a delimiter d is_found where d ∈
  (pattern_after|stop_word|punctuation|title_start)
    Else
  Check for names from w_i+1in the same cluster;
  Stop when a delimiter d is_found where d ∈
  (pattern_after|stop_word|punctuation|title_start)
  Else if w_i in honorific_list
  Check for names from w_i+1 in the same cluster;
    Stop when a delimiter d is_found where d ∈
(pattern_after|stop_word|punctuation|title_start)
```

The above rule is used to extract names from sentences such as:

قال *الرئيس* محمد مرسي **ان** مصر تخطو ...

President Mohammad Morsi said that Egypt is stepping through …

اكد *الدكتور* محمد حجازي *مساعد* وزير الخارجية ...

Dr. Mohammad Higazy, Deputy of minister of foreign affairs confirmed that …

قال وليد جنبلاط *رئيس* كتلة اللقاء ...

Walid Junblatt the president of … said…

This rule is generalized to extract names from sentences which contain multi honorifics before the person's name such as:

قال *رئيس الوزراء الاسرائيلي* ايهود اولمرت **إنه** عازم ...

Prime Minister of Israel Ehud Olmert said that he will …

Another rule is used to check for a pattern followed by an unknown name (not in the dictionaries) with the prefix عبد (Abd) followed by known male nameand/or fami-

[6] http://www1.ccls.columbia.edu/~ybenajiba/downloads.html

ly name (the previous stopping criterion is used). Also we utilize the fact that some names could appear with the conjunction و (and), hence a rule is used to extract a known first name preceded by و (and) followed by known male name and/or family name(the previous stopping criterion is used).Other rules are employed, but are not included due to space limitations.

3 System Evaluation

The presented system was evaluated using the precision, recall and f-score measures based on what it extracted as names from the benchmark ANERcorp dataset. Table 2 provides a comparison between the results of the presented system with two state of the art systems which are the hybrid NERA approach [1] and the machine learning approach using conditional random fields (CRF) [10].

Table 2. Comparison between our system performance in terms of precision, recall and F-score with the current two state of the art systems

	Precision	Recall	F-score
Hybrid System	94.9	90.78	92.8
CRF System	80.41	67.42	73.35
Our System	**92.29**	**72.75**	**81.36**

From this comparison, it can be inferred that our system competes with the state of the art systems using the introduced novel approach in terms of precision. However the recall of our system is still below the recall of the state of the art hybrid approach. One of the reasons that decrease the recall is the fact that our main rule is to extract full name which is composed of a first name followed by other male names and/or a family name, in order to avoid the effects of false positives. So we did not extract single names such as ميسي (Messi) and رونالدينيو (Ronaldinio) in the phrase:

... لكن نجميه رونالدينيو والارجنتيني ميسي اضاعا...

…but its stars Ronaldinio and the argentine Messi have missed …

Another reason is that, using our dictionaries in addition to the above rule, a name such as ابنيزر نجوي (Ebenezer Najwa) could not be extracted, because according to our dictionaries (Najwa) is a first name so it should not be found after (Ebenezer).

To handle the issue of the relatively low recall, we will try to make the used rules more flexible while keeping an acceptable precision value.

Table 3 shows the effect of using clusters in boosting the system precision besides the final system results using patterns and clusters. The use of clusters decreases the effect of false positives so the precision increased by 6points.

Table 3. Individual system‘s components evaluation vs. the complete system in terms of precision

	Precision	Recall	F-score
Dictionaries Only	71.0	62.98	66.75
Dictionaries+Clusters	77.24	58.62	66.65
Dictionaries+Clusters+Patterns	**92.29**	**72.75**	**81.36**

4 Related Work

NER systems for Arabic can be classified based on the type of text being processed; whether it is formal Modern Standard Arabic (MSA) or informal colloquial Arabic text. Then they can be further classified based on the used approach, whether it is rule based, machine learning based or hybrid.

The majority of previous work addressing NER in Arabic language was developed for formal MSA text which is the literary language used in newspapers and scientific books. However the informal colloquial Arabic, corresponding to the spoken dialectic, is currently being used widely in social media communication.

As mentioned earlier, rule based NER based on handcrafted rules combined with gazetteers. One of the initial systems which combined a generic pattern matching engine with high-precision morphological text analysis to recognize Arabic named entities was TAGARAB [17]. TAGARAB depends on a morphological analysis module plugged into a morphological tokenizer in addition to lists of nouns, verbs, and adjective stems to partially support a series of regular expressions.

Another technique was presented to extract proper names from Arabic text for a question-answering system [3]. The technique depends on collecting information about the words in the text and building graphs to represent the relationships between them.

A system based on local grammar (patterns) to extract persons' names from Arabic news articles is described in [24]. The used approach is based on the fact that persons' names cluster around certain frequent verbs in news articles. Collocation analysis is done to discover the words that frequently collocate with the Arabic reporting verbs such as prepositions, punctuations and function words. Then concordances analysis is generated to return frequency information and citations for the searched reporting verbs and each of its inflected forms. Finally a Finite State Automata (FSA) is constructed for the reporting verbs extracted patterns.

PERA [21] is a system for extracting Arabic persons' names. The system adopts a rule based approach using linguistic grammar-based techniques. Grammar rules, supported by gazetteers, were built based on keywords or trigger words to form a window around a person's name. PERA was evaluated on purpose-built corpora using ACE and Treebank news corpora that were tagged in a semi-automated way. The system has been generalized as NERA [22, 23] to extract other named entities.

The work presented in [15] describes a person named entity recognition system for the Arabic language. The system makes use of heuristics to identify person names and is composed of two main parts: the General Architecture for Text Engineering (GATE) environment and the Buckwalter Arabic Morphological Analyzer (BAMA). The system makes use of a huge set of dictionaries. The same work was repeated in [16] and compared with [21] and [19].

As mentioned in [1], the frequently used approach for NER is the machine learning approach by which text features are used to classify the input text depending on an annotated dataset.

Benajiba et al. applied different machine learning techniques [7-12] to extract named entities from Arabic text. The best performing of these makes use of optimized feature sets [10].

ANERSys [7] was initially developed based on n-grams and a maximum entropy classifier. The maximum entropy classifier basically computes for each word the probability that it will be assigned to each of the considered classes using the maximum entropy formula and then assigns the class with the highest probability to this word. Moreover a training and test corpora (ANERcorp) and gazetteers (ANERgazet) were developed to train, evaluate and boost the implemented technique. ANERcorp is currently considered the benchmark dataset for testing and evaluating NER systems.

ANERSys 2.0 [8] basically improves the initial technique used in ANERSys by combining the maximum entropy with POS tags information. Hence the recognition of long named entities is improved by extracting the boundaries of the named entity.

By changing the probabilistic model from Maximum Entropy to Conditional Random Fields the accuracy of ANERSys is enhanced [9]. Another system is introduced which makes use of leading and trailing character n-grams in words in addition to other surface and word association features to train a conditional random field's model [2].

A novel approach is described in [11] to extract Arabic named entities using Support Vector Machines with the aid of contextual, lexical and morphological features combination.

A recent attempt to extract named entities from Arabic text using an artificial neural network is discussed in [20]. The system uses a back propagation training algorithm in addition to selecting an appropriate set of features for each named entity class.

Hybrid approaches combine machine learning techniques, statistical methods and predefined rules. In [4] a hybrid system built based on both statistical methods and predefined rules to extract Arabic named entities, is described. The system combines three different techniques: rules, graphs, and statistics. Rules are used to mark named entities phrases. A graph-based method is implemented to represent the relationships between words. Finally rules and the frequency of tokens are utilized to identify proper names.

The most recent hybrid NER system for Arabic uses a rule based NER component integrated with a machine learning classifier [1]. The system operates over two stages. In the first stage, a re-implementation of the NERA system [22, 23] using the GATE platform, is used to tag the words of the input text. In the second stage, the outputs of the rule based system are propagated as features to a decision-tree machine learning classifier along with other general features. The Stanford POS Tagger has been used to compute some of these other features, such as word category and affixation. The reported results of the system are significantly better than the pure rule based system and the pure machine learning classifier. In addition the results are also better than the state of the art Arabic NER system based on conditional random fields [10].

4.1 Differences Between our System and Previous Work

From the previous discussion, it can then be inferred that, the currently used rule based approaches to extract named entities from MSA text, are dependent on tokenizers, taggers and parsers combined with a huge set of gazetteers. Although, those approaches might be sufficient for extracting persons' names from a formal domain, it will be hard to modify them for the informal "colloquial Arabic" domain. Similarly, machine learning approaches make use of taggers, parsers and set of gazetteers to extract contextual, lexical and morphological features. Those features are used to train different classifiers. In addition, an annotated corpus is always required for training.

Our approach, which combines a rule based approach with a statistical one, avoids the use of parsers, taggers and morphological analyzers. All the system requires is a large set of names, which can be easily obtained from public resources such as Wikipedia. The main challenges addressed by this work are to overcome the ambiguity problem of persons' names, to avoid the shortcomings of both the rule based NER and the machine learning based NER approaches and to build a domain independent persons' names extraction system.

There is some similarity between our approach and the one based on local grammar [24] as later uses reporting verbs as patterns to indicate the occurrence of persons' names. However our approach extracts patterns automatically from the same domain under study, so the patterns are not limited to a list of reporting verbs.

There was an attempt to recognize named entities from documents written in Indonesian language using association rules [14]. The system uses association rules in terms of support and confidence to extract named entities. A set of previously defined features, dictionaries and name classes from an annotated corpus is employed to describe the two sets of items from the dataset which are used to calculate the support. One of those sets is the name of the class to be predicted and the other is all the possible forms of the class, but the overall approach taken by that system is different than the one we presented.

5 Conclusion

This paper presented a novel approach for extracting persons' names from Arabic text. This approach integrated name dictionaries and name clusters with a statistical model based for extracting patterns that indicate the occurrence of persons' names. The used approach overcomes major limitations of the rule based approach which are the need of huge set of gazetteers and domain dependence. Using this system, persons' name extraction could be applied on new domains without facing difficulties to import the system into the new domain. Our rule based approach was able to overcome the ambiguity of Arabic persons' names using clusters besides the original dictionaries of names. Building the patterns' statistical model using association rules improved the tasks of Arabic persons' names disambiguation and extraction from any domain. System evaluation, on the benchmark dataset, showed that the performance of the presented technique is comparable to the state of the art machine learning ap-

proach. However, it still needs some improvements to compete with the state of the art hybrid approach.

References

1. Abdallah, S., Shaalan, K.,Shoaib, M.: Integrating rule-based system with classification for Arabic named entity recognition. In: Gelbukh, A. (ed.) CICLing 2012. LNCS, vol. 7181, pp. 311–322. Springer, Heidelberg (2012)
2. Abdul Hamid, A., Darwish, K.: Simplified feature set for Arabic named entity recognition. In: Proceedings of the 2010 Named Entities Workshop, pp. 110–115. Association for Computational Linguistics, Uppsala (2010)
3. Abuleil, S.: Extracting names from Arabic text for question-answering systems. In: Proceedings of Coupling approaches, coupling media and coupling languages for information retrieval, RIAO 2004,pp. 638-647. Avignon(2004)
4. Abuleil, S.: Hybrid system for extracting and classifying Arabic proper names. In: Proceedings of the fifth WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED 2006, pp. 205-210. Madrid (2006)
5. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, SIGMOD 1993, pp. 207–216. Washington (1993)
6. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, pp.361–362. California (2009)
7. Benajiba, Y., Rosso, P., Benedì Ruiz, J.M.: ANERsys: An Arabic named entity recognition system based on maximum entropy. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007)
8. Benajiba, Y., Rosso, P.: Anersys 2.0: Conquering the ner task for the Arabic language by combining the maximum entropy with pos-tag information. In: IICAI, pp. 1814–1823 (2007)
9. Benajiba, Y., Rosso, P.: Arabic named entity recognition using conditional random fields. In: Workshop on HLT & NLP within the Arabic World. Arabic Language and Local Languages Processing: Status Updates and Prospects (2008)
10. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition using optimized feature sets. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 284–293. Association for Computational Linguistics, Morristown (2008)
11. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition: An svm-based approach. In: The International Arab Conference on Information Technology, ACIT 2008 (2008)
12. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition: A feature-driven study. IEEE Transactions on Audio, Speech, and Language Processing 17(5), 926–934. (2009)
13. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment p10008, (2008)
14. Budi, I., Bressan, S.: Association rules mining for name entity recognition. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE 2003, pp. 325–336. Italy (2003)

15. Elsebai, A., Meziane, F., Belkredim, F. Z.: A rule based persons names Arabic extraction system. Communications of the IBIMA 11, 53–59.(2009)
16. Elsebai, A., Meziane, F.: Extracting person names from Arabic newspapers. In: Proceedings of the International Conference on Innovations in Information Technology, IIT 2011, pp.87–89.UAE (2011)
17. Maloney, J., Niv, M.: TAGARAB: a fast, accurate Arabic name recognizer using high-precision morphological analysis. In: Proceedings of the Workshop on Computational Approaches to Semitic Languages, Semitic 1998, pp. 8–15. Association for Computational Linguistics, Morristown (1998)
18. Mansouri, A., Affendey, L.S., Mamat, A.: Named entity recognition using a new fuzzy support vector machine. In: Proceedings of the 2008 International Conference on Computer Science and Information Technology, ICCSIT 2008, pp. 24–28. Singapore (2008)
19. Mesfar, S.: Named entity recognition for Arabic using syntactic grammars. In: NLDB 2007, LNCS, vol. 4592, pp. 305-316. Springer, Heidelberg (2007)
20. Mohammed, N.F., Omar, N.: Arabic named entity recognition using artificial neural network. Journal of Computer Science 8(8), 1285-1293. (2012)
21. Shaalan, K., Raza, H.: Person name entity recognition for Arabic. In: Fifth Workshop on Important Unresolved Matters, pp. 17–24.Czech Republic(2007)
22. Shaalan, K., Raza, H.: Arabic Named Entity Recognition from Diverse Text Types. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 440–451. Springer, Heidelberg (2008)
23. Shaalan, K., Raza, H.: NERA: Named entity recognition for Arabic. Journal of the American Society for Information Science and Technology, 1652–1663 (2009)
24. Traboulsi, H.: Arabic named entity extraction: A local grammar-based approach. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 4, pp. 139–143 (2009)
25. Zayed, O.H., El-Beltagy, S.R.: Person Name Extraction from Modern Standard Arabic or Colloquial Text. In: Proceedings of the eighth International Conference on Informatics and Systems, INFOS 2012, pp. NLP-44–NLP-48.Egypt (2012)

A Symbolic Corpus-based Approach to Detect and Solve the Ambiguity of Discourse Markers

Iria da Cunha

University Institute for Applied Linguistics
Universitat Pompeu Fabra
C/ Roc Boronat, 138, 08018, Barcelona, Spain
iria.dacunha@upf.edu

Abstract. At present, discourse parsing is an important research topic. Rhetorical Structure Theory (RST) is one of the most popular approaches in this field. In general, discourse parsing includes three stages: discourse segmentation, discourse relations detection and building up rhetorical trees. Different strategies are used when developing discourse parsers. One of the strategies to detect discourse relations is based on symbolic rules that take into account linguistic clues, such as discourse markers. Nevertheless, some discourse markers are ambiguous, that is, they can indicate more than one discourse relation. This fact constitutes a problem when assigning discourse relations automatically. In this paper, a symbolic approach to detect and solve discourse markers ambiguity in Spanish is developed. First, we detect ambiguous discourse markers, using the training corpus of the RST Spanish Treebank. Second, we extract linguistic contexts for these markers. Third, we design linguistic rules to solve the ambiguity of discourse markers. Fourth, we evaluate the rules, using the test corpus of the RST Spanish Treebank. Our approach outperforms the baseline created following the methodology of the state of the art. Therefore, we consider that the results obtained in our experiments are representative and constitute the first step towards the disambiguation of discourse markers senses in Spanish. However, there is room for improvement and the main limitations of the approach are presented. In the future, the rules will be integrated in a discourse parser for Spanish, and several related applications will be developed (automatic summarization and information extraction, among others).

Keywords: Discourse Parsing, Discourse Markers, Ambiguity, Corpus, Rhetorical Structure Theory, Spanish

1 Introduction[1]

At present, discourse parsing is an important research topic, since it is being widely used to develop several applications, such as automatic summarization, information extraction, text generation, automatic translation, sentence compression, coherence evaluation, etc. Rhetorical Structure Theory (RST) [1] is one of the most popular

[1] This work has been financed by the Spanish projects RICOTERM 4 (FFI2010-21365-C03-01) and APLE 2 (FFI2012-37260), and a *Juan de la Cierva* grant (JCI-2011-09665).

approaches in this field. RST is a language-independent theory based on the idea that a text can be segmented into Elementary Discourse Units (EDUs) linked by means of nucleus-satellite or multinuclear rhetorical relations. In the first case, the satellite gives additional information about the other unit (the nucleus), on which it depends (e.g. Cause, Purpose or Result). In the second case, several elements, all nuclei, are connected at the same level, i.e. there are no dependent elements and they all are equally important with regard to the author's intentions (e.g. List, Contrast or Sequence). In general, discourse parsing includes three stages: discourse segmentation, discourse relations detection and building up rhetorical trees. RST-based discourse parsers for some languages are available: English [2], [3], [4], Japanese [5], Brazilian Portuguese [6] and Spanish [7]. These parsers use symbolic or statistical approaches. One of the strategies to detect discourse relations is based on symbolic rules which take into account linguistic clues, such as discourse markers [6], [8]. Traditionally, discourse markers are defined as invariable linguistic units that guide inferences in communication (see [9] for a review on discourse markers definitions). However, as [9] mentions: "[…] the signalling of discourse relations is not restricted to discourse markers; many other devices are used to signal the presence of such relations". Thus, we do not follow the traditional definition of discourse markers, but we use this term in a wide sense.

It is important to highlight that some discourse markers are ambiguous. Specifically, as [10] state:

> There are two types of ambiguity that need to be resolved during discourse processing. First, a word can be ambiguous between discourse or non-discourse usage. For example, 'once' can be either a temporal discourse connective or a simply a word meaning 'formerly'. Secondly, some connectives are ambiguous in terms of the relation they mark. For example "since" can serve as either a temporal or causal connective.

In this work, we focus on the second type of ambiguity. As [11] states, one of the problems of the semantics of natural connectors is that the same connector can express different connection types and one connection type can be expressed by several connectors. For example, the Spanish marker *mientras* ("while") can express at the same time Contrast, Circumstance and Condition. Regarding discourse parsing, [2] highlights that discourse markers can indicate more than one discourse relation and this fact constitutes a problem when assigning discourse relations automatically. When working on discourse parsing (specifically, in the case of automatic relation detection), three strategies can be used to deal with the problem of markers ambiguity: a) to choose the relation with a higher number of markers of this type, b) to give to the algorithm all possible relations, or c) to develop more fine-grained strategies combining several markers to choose only one relation.

The main objectives of this paper are: a) to detect ambiguous discourse markers in Spanish, and b) to develop fine-grained strategies in order to solve the ambiguity of discourse markers automatically. This is the first study that aims at detecting and solving the ambiguity of discourse markers in Spanish, considering ambiguity as the possibility to indicate more than one discourse relation.

In Section 2, related work is presented. In Section 3, the methodology used in the study is explained. In Section 4, corpus analysis and results are presented. In Section 5, the evaluation of the results is shown. In Section 6, some conclusions and future work are established.

2 Related Work

Most of the work on discourse markers disambiguation has been done for English. [12] carry out an empirical study on discourse and sentential uses of cue phrases, in which text-based and prosodic features are examined for disambiguation power. They propose that discourse or sentential usage can be distinguished by intonational features, and present a prosodic model that characterizes these distinctions. [13] present a set of manual sense annotation studies for three connectives in English ("since", "while" and "then"), whose arguments have been annotated in the Penn Discourse Treebank (PDTB) [14]. They use syntactic features annotated in this corpus and a maximum entropy model to automatically disambiguate the sense of these connectives. In this work only three specific connectors are analyzed, the corpus contains texts written in English, and machine learning is used. Nevertheless, we consider that some of the used features are interesting (mainly verbal tense), and we use them in our work. [15] also mention the difficulty of disambiguating discourse markers senses, especially when classifying rhetorical relations automatically. They propose a supervised machine learning method that uses several linguistic features to classify discourse relations in the absence of a cue phrase. They introduce the idea that tense and aspect offer clues about temporal relations and could influence the probabilities of different rhetorical relations. [10] use a Naive Bayes classifier to demonstrate that syntactic features improve performance in both discourse and non-discourse disambiguation tasks. In their experiments, they consider only the four top categories in the PDTB (Expansion, Comparison, Contingency and Temporal), obtaining a high accuracy in both experiments. Nevertheless, they do not offer a list and a linguistic analysis of the markers they use.

Some work on this subject is also found for other languages, such as German [16], [17] and Arabic [18]. For Spanish, there are few studies. We highlight the work of [19], who presents a proposal for detection and classification of Spanish discourse markers. Nevertheless, he mainly deals with the first type of ambiguity (sentence vs. discourse use of markers). In this study, punctuation (mainly the comma) is used to disambiguate discourse markers function. Later, [20] uses this work to create a system for detecting Spanish discourse markers automatically, but the main feature to identify different meanings is again punctuation.

After the revision of the state of the art, we can draw some conclusions: a) the disambiguation of discourse markers senses is a language-dependent task, since the lexical, syntactic and discourse features differ among languages; b) there is a research gap on this subject in Spanish; c) Spanish is a language with a high degree of syntactic complexity, and explicit Spanish discourse markers are more ambiguous than English markers, so the disambiguation task in Spanish is challenging; and d) to our knowledge, there are no studies carrying out a corpus analysis to detect the most frequent ambiguous markers in a language, or observing linguistic regularities in the different discourse relations they show.

3 Methodology

In the first stage, we use the database of Spanish discourse markers and RST relations proposed by [7] *to extract ambiguous discourse markers*. In other words, we extract discourse markers signalling more than one relation in the database. This database was created using the training corpus of the RST Spanish Treebank [21], which includes texts annotated with rhetorical relations (http://corpus.iingen.unam.mx/rst/). The corpus contains texts from nine specialized domains (Astrophysics, Earthquake Engineering, Economy, Law, Linguistics, Mathematics, Medicine, Psychology and Sexuality) and several genres (research articles, abstracts, sections of manuals and books, etc.). This variety of domains and genres guarantees that the results can be generalized. The corpus is divided into training corpus (183 texts) and test corpus (84 texts). It includes 52,746 words, 267 texts, 2,256 sentences and 3,349 discourse segments. The database mentioned above includes three types of markers:

1. Traditional discourse markers, such as *ya que* ("since").
2. Markers including lexical units, specifically, nouns and verbs, such as *metodología* ("methodology").
3. Markers including verbal structures, such as *para* ("to") + infinitive.

In our work, linguistic markers of Elaboration relations are not analyzed, since this is the most general and frequent relation in the language. We detect 31 markers indicating more than one relation in the training corpus. Over this list of ambiguous markers, two filters are applied: a) only the first and third types of discourse markers are analyzed, and b) only discourse markers with a frequency higher than the one in the corpus are taken into account. Thus, we obtain the 11 ambiguous discourse markers to be analyzed. Table 1 includes these markers and the marked relations (with their frequency in the training corpus indicated in brackets).

Table 1. Ambiguous discourse markers found in the corpus and analyzed in this work

Marker	Marked relations
pues ("since")	Cause (4), Justification (5)
ya que ("because")	Cause (2), Justification (3)
debido a ("due to")	Cause (7), Justification (4)
mientras ("while")	Contrast (11), Circumstance (2)
después ("after")	Sequence (3), Circumstance (4)
cuando ("when")	Condition (5), Circumstance (22)
y ("and")	Contrast (3), List (11)
o ("or")	Disjunction (6), Contrast (3)
al ("when") + infinitive	Cause (2), Circumstance (10)
comma + *lo que* ("which")	Interpretation (5), Result (6)
gerund verbal form	Concession (1), Condition (2), Result (7), Means (8), Circumstance (16)

In the second stage, *the discourse contexts of these ambiguous markers are extracted*, by using the RST_extract tool [22], which offers to the user text passages corresponding to discourse relations. We consider contexts to be: a) two EDUs (Nucleus and Satellite) in nucleus-satellite relations, and b) several EDUs (Nuclei) in multinuclear relations.

In the third stage, the contexts are analyzed automatically by using Freeling syntactic parser [23], available at: http://nlp.lsi.upc.edu/freeling/. Then, the contexts of each marker are analyzed manually, in order to find linguistic regularities in the contexts of each RST relation. These regularities are used *to develop rules capable of disambiguating the discourse markers senses*, that is, detecting the discourse relation they are marking in a specific context. The features that we analyze are:

- Verbal tense and mode (such as present vs. past, or indicative vs. subjunctive).
- Verbal lexical units (such as "to use", "to consider", etc.).
- Affirmative vs. negative verbal form.
- Position of the marker (such as at the beginning of the EDU).
- Combination of markers (such as "and + while").
- Subjects of the related EDUs.
- Punctuation (such as the comma).

In the fourth stage, *the developed rules are evaluated*, using the test corpus of the RST Spanish Treebank.

4 Analysis and Results

After analyzing the contexts of ambiguous discourse markers and detecting regularities, the disambiguation rules are designed and a template is created for each discourse marker (see Tables 2-10). In this corpus analysis, we observe different regularities, which are explained in this section.

I) The markers *pues* ("then", "since") and *ya que* ("because", "since") can express the relation of Justification or Cause. To justify an idea, speakers commonly use several arguments or related statements; therefore, sentences including a relation of Justification tend to contain several EDUs (usually with various discourse markers). On the contrary, to express the relation of Cause, speakers usually offer a fact first and then the cause of this fact directly, so the sentence includes only two EDUs (see Table 2). For example:

[*Los estudiantes adultos de origen chino, coreano y japonés tienen problemas para pronunciar los fonemas líquidos del españo*l]NUCLEUS [***ya que*** *en su lengua hay un solo fonema para estos sonidos.*]SATELLITE_CAUSE

[The adult students of Chinese, Korean and Japanese origin have problems to pronounce the liquid phonemes of the Spanish] [**since** in their language there is a single phoneme for these sounds.]

Table 2. Rule template for the markers *pues* ("then", "since") and *ya que* ("because", "since")

Discourse marker	Disambiguation rule
pues ("since") *ya que* ("because") ----------	IF 2 EDUs are related by the discourse marker *pues* ("then", "since") OR *ya que* ("because", "since") AND the 2 EDUs are included in a sentence consisting of only 2 EDUs
Marked relations	THEN relation = Cause
Cause	ELSE IF the 2 EDUs are included in a sentence consisting of more than 2 EDUs
Justification	THEN relation = Justification

In the case of the marker *debido a* ("due to"), it has not been possible to design a disambiguation rule. The regularities detected in the contexts of the markers *pues* and *ya que* have not been observed in the contexts of this marker, which can also express

Justification or Cause. Due to the lack of examples (two cases of Justification and three cases of Cause) we do not have enough information available, and more cases would be necessary in order to elaborate an adequate rule.

II) The marker *mientras* ("while") can signal the relation of Contrast or Circumstance. On the one hand, when making a Contrast between two elements, something is being argued or compared. On the other hand, the relation of Circumstance only offers some information or data (see Table 3). For example:

[***Mientras*** *se preparan dichas herramientas,*]SATELLITE_CIRCUMSTANCE [*habremos de trabajar sobre la modelización de los términos técnicos.*]NUCLEUS

[**While** these tools are prepared,] [we will have to work on the modelization of the technical terms.]

Table 3. Rule template for the marker *mientras* ("while")

Discourse marker	**Disambiguation rule**
mientras ("while") ---------- **Marked relations** Contrast Circumstance	@mientras = {*ya que* ("because, since"), *pues* ("then, since"), *por un/otro lado* ("on the one/other hand"), *por este/ese/aquel* ("on this/that case"), *en el primer/segundo/tercero/cuarto caso/lugar* ("in the first/second/third/fourth case/place"), *en este/ese/aquel caso* ("in this/that case")} IF 2 EDUs are related by the discourse marker *mientras* ("while") AND the marker is followed by the conjunction *que* ("that") OR the marker appears in combination with another discourse marker in @mientras THEN relation = Contrast ELSE IF the marker is not combined with *que* or another discourse marker in @mientras THEN relation = Circumstance

III) An EDU starting with the discourse marker *después* ("after") can be a part of a Sequence or indicate a Circumstance. On the one hand, if this marker appears in a segment constituting a single sentence, the relation should be Sequence, since the content is not offering a Circumstance of another segment. On the other hand, if the marker relates two segments in the same sentence, it could indicate a Circumstance (if the structure [después + *de* ("of") + infinitive] appears) or Sequence (if some other structure appears) (see Table 4). For example:

[*El virus se multiplica en las células y en la base de la lesión,*]NUCLEUS [*e infecta la neurona que los inerva (ganglio sacro).*]NUCLEUS [***Después*** *el virus volverá al punto inicial.*]NUCLEUS_SEQUENCE

[The virus is multiplied in the cells and in the base of the injury,] [and infects the neuron that innervates them (sacred ganglion).] [**Afterwards** the virus will return to the initial point.]

Table4. Rule template for the marker *después* ("after")

Discourse marker	**Disambiguation rule**
después ("after") ---------- **Marked relations** Sequence Circumstance	IF 2 EDUs are related by the discourse marker *después* ("after") AND the 2 EDUs are included in different sentences THEN relation = Sequence ELSE IF the 2 EDUs are included in the same sentence AND after *después* the preposition *de* ("of") appears, followed by an infinitive verbal form THEN relation = Circumstance ELSE IF the 2 EDUs are included in the same sentence AND the discourse marker is not followed by the structure *después* + *de* + infinitive verbal form THEN relation = Sequence

IV) The discourse marker *cuando* ("when") can indicate the relation of Circumstance or Condition. To determine which one of these two senses is correct, it is necessary to analyze the verbal tense and/or mode of the two EDUs that the marker relates. For example, if the main verb of the EDU containing the marker is a subjunctive verbal form, and the main verb of the other EDU is a present or future verbal form, the relation should be Condition; however, if the main verbs of the two EDUs are past forms, the relation should be Circumstance (see Table 5). For example:

[***Cuando*** *entramos a la sala de exhibición*]SATELLITE_CIRCUMSTANCE [*el susto fue inmenso.*]NUCLEUS

[**When** we enter in the exhibition room] [the fright was immense.]

Table 5. Rule template for the marker *cuando* ("when")

Discourse marker	Disambiguation rule
cuando ("when")	IF 2 EDUs are related by the discourse marker *cuando* ("when")
----------	AND the main verb of the EDU including the marker is a past verbal form
Marked relations	AND the main verb of the EDU not including the marker is a past verbal form
Condition	THEN relation = Circumstance
Circumstance	ELSE IF the main verb of the EDU including the marker is a present verbal form
	AND the main verb of the EDU not including the marker is a gerund verbal form
	THEN relation = Circumstance
	ELSE IF the main verb of the EDU including the marker is a subjunctive verbal form
	AND the main verb of the EDU not including the marker is a present OR future verbal form
	THEN relation = Condition
	ELSE IF the main verb of the EDU including the marker is a reflexive present verbal form
	AND the main verb of the EDU not including the marker is a present verbal form
	THEN relation = Circumstance
	ELSE IF the main verb of EDU including the marker is a non-reflexive present verbal form
	AND the main verb of the EDU not including the marker is a present verbal form
	THEN relation = Condition

V) Usually, in Spanish, the marker *y* ("and") marks the end of a List. Nevertheless, if this marker appears combined with another negation marker, it can indicate Contrast (see Table 6). For example:

[*No vulnera el sistema constitucional ni en general el orden jurídico*]NUCLEUS [***y sí, en cambio****, asegura que los derechos de la persona sean mejor protegidos y garantizados.*]NUCLEUS_CONTRAST

[It does not interfere in the legal order, neither in general in the constitutional system] [**and, by contrast,** it guarantees that people rights are better protected and guaranteed.]

Table 6. Rule template for the marker *y* ("and")

Discourse marker	Disambiguation rule
y ("and")	@contrast = {*no* ("no"), *en cambio* ("on the other hand"), *por el contrario* ("by contrast") *otro/otros/otra/otros* ("another/other/others")}

Marked relations	IF 2 EDUs are related by the discourse marker *y* ("and")
Contrast	AND *y* is combined with another discourse marker in @contrast
List	THEN relation = Contrast
	ELSE IF *y* is not combined with another discourse marker in @contrast
	THEN relation = List

VI) The marker *o* ("or") can signal the relation of Disjunction or Contrast. In the first case, the related EDUs have the same subjects while, in the second case, the subjects are different (see Table 7). For example:

[*¿Son términos todos los que lo parecen*]NUCLEUS **[*o abundan las creaciones léxicas sensacionalistas y efímeras?*]**NUCLEUS_CONTRAST

[Are all those that seem it terms] **[or** do the sensationalist and ephemeral lexical creations abound?]

Table 7. Rule template for the marker *o* ("or")

Discourse marker	Disambiguation rule
o ("or")	IF 2 EDUs are related by the discourse marker *o* ("or")
----------	AND the 2 EDUs have the same subject
Marked relations	THEN relation = Disjunction
Disjunction	ELSE IF the 2 EDUs have not the same subject
Contrast	THEN relation = Contrast

VII) In Spanish the construction [*al* ("when", "as") + infinitive] can be used to indicate a Cause or a Circumstance discourse relation (see Table 8). In the corpus, negative cases indicate Cause. For example:

[***Al no contar*** *en Cuba con propias referencias acerca del desarrollo del lenguaje infantil,*]SATELLITE_CAUSE [*se realizó una investigación nacional descriptiva y transversal.*]NUCLEUS

[**As** Cuba **does not have** its own references about the development of the infantile language,] [a descriptive and transversal national research was carried out.]

Table 8. Rule template for the marker *al* ("when", "as") + infinitive

Discourse marker	Disambiguation rule
al ("when") +	IF 2 EDUs are related by the syntactic construction *al* ("when", "as") + infinitive
infinitive	AND the construction includes a negation
----------	THEN relation = Cause
Marked relations	ELSE IF the 2 EDUs does not include a negation
Cause	THEN relation = Circumstance
Circumstance	

VIII) If an EDU starts with the relative pronoun *lo que* ("which") preceded by a comma, it can express the relation of Result or Interpretation. In this case, in order to differentiate both senses, the verb included in the EDU is used, since, in general, speakers use different verbs to express an objective result or their interpretation about something (e.g. *causar* ["to cause"] vs. *suponer* ["to supposse"]). For example:

[*Durante la pubertad, los niveles elevados de estrógenos hacen que el epitelio vaginal se adelgace y que el contenido de glucógeno celular se incremente,*]NUCLEUS [*lo que* ***provoca*** *que el pH vaginal disminuya.*]SATELLITE_RESULT

[During the puberty, the high levels of estrogens make the vaginal epithelium lose weight and the contents of cellular glycogen be increased,] [which **causes** that the vaginal pH decreases.]

Table 9. Rule template for the marker *lo que* ("which") preceded by a comma

Discourse marker	Disambiguation rule
lo que ("which") preceded by a comma --- **Marked relations** Interpretation Result	@interpretation = {*permitir* ("to allow"), *poner de manifiesto* ("to show"), *suponer* ("to suppose"), *conllevar* ("to entail")} @result = {*agudizar* ("to aggravate"), *causar* ("to cause"), *complicar* ("to complicate"), *conducir a* ("to lead to"), *dar lugar* ("to give place to"), *generar* ("to generate"), *hacer que* ("to cause"), *llegar* ("to arrive"), *manifestarse* ("to appear"), *obtener* ("to obtain"), *ofrecer* ("to offer"), *propiciar* ("to favour"), *provocar* ("to cause"), *resultar* ("to result"), *ser utilizado* ("to be used")} IF 2 EDUs are related by the relative *lo que* ("which") preceded by a comma AND the main verb of the EDU containing the relative is included in @result THEN relation = Result ELSE IF the main verb of the EDU containing the relative is included in @interpretation THEN relation = Interpretation

IX) In Spanish, gerund verbal forms should be used only to indicate simultaneity. Nevertheless, probably due to the influence of English, Spanish speakers tend to use gerunds to indicate Result, Concession, Means or Circumstance discourse relations. For example:

[*El objetivo de este trabajo es analizar los efectos de la política monetaria en el producto y los precios en la economía mexicana*]NUCLEUS [***utilizando*** *diversas técnicas econométricas.*]SATELLITE_MEANS

[The goal of this work is to analyze the effects of the currency policy in the product and the prices in the Mexican economy] [**using** different econometric techniques.]

Table 10. Rule template for the marker gerund verbal form

Discourse marker	Disambiguation rule
gerund verbal form --- **Marked relations** Concession Condition Result Means Circumstance	@result = {*agudizar* ("to aggravate"), *causar* ("to cause"), *complicar* ("to complicate"), *conducir a* ("to lead to"), *dar lugar* ("to give place to"), *generar* ("to generate"), *hacer que* ("to cause"), *llegar* ("to arrive"), *manifestarse* ("to appear"), *obtener* ("to obtain"), *ofrecer* ("to offer"), *propiciar* ("to favour"), *provocar* ("to cause"), *resultar* ("to result"), *ser utilizado* ("to be used")} @means = {*advertir* ("to advise"), *aplicar* ("to apply"), *aprovechar* ("to benefit"), *basarse* ("to be based on"), *comparar* ("to compare"), *controlar* ("to control"), *emplear* ("to use"), *esquematizar* ("to outline"), *estudiar* ("to study"), *hacer uso* ("to use"), *incluir* ("to include"), *incorporar* ("to incorporate"), *indagar* ("to investigate"), *plantear* ("to lay out"), *seguir* ("to continue"), *seleccionar* ("to select"), *tomar como base* ("to take as a base"), *tomar en cuenta* ("to take into account"), *trabajar* ("to work"), *usar* ("to use"), *utilizar* ("to use")} IF 2 EDUs are related by a gerund verbal form AND the gerund is preceded by the marker *aun* ("even") THEN relation = Concession ELSE IF the gerund is included in the EDU placed in the first position of the sentence THEN relation = Condition ELSE IF the gerund is not included in the EDU placed in the first position of the sentence AND the gerund is a verb included in @result THEN relation = Result AND the gerund is a verb included in the @means THEN relation = Means AND the gerund is not a verb included in @result OR @means THEN relation = Circumstance

Our disambiguation approach includes eight rules. These rules take into account all the features analyzed, except the punctuation feature. Unlike [19], we do not find that the comma offers relevant information in order to disambiguate discourse senses of Spanish markers. We consider that the reason is that, in Spanish, there are many cases in which the use of the comma is optional. Maybe this feature can help to differentiate between sentential and discourse uses of markers, but it is not useful to differentiate between their different discourse meanings.

5 Evaluation

In order to evaluate the performance of our approach, we use the test corpus of the RST Spanish Treebank, which constitutes a gold standard for Spanish. This corpus includes 84 texts, from the Mathematics, Psychology and Sexuality domains. Using again the RST_extract tool, we extract from this test corpus contexts with the following characteristics: a) the context includes an ambiguous discourse marker of our database; b) the context corresponds to one RST relation that can be expressed by that discourse marker, and c) a disambiguation rule has been created for this marker. 61 contexts are obtained.

Then, the disambiguation rules are applied to each context, in order to detect the RST relation that the contexts include. When applying the disambiguation rules, we assume that the EDUs related by the marker are previously detected. We obtain them by using the discourse segmenter DiSeg [24], available at: http://daniel.iut.univ-metz.fr/DiSeg/WebDiSeg/.

We calculate the number of contexts including ambiguous discourse markers that have been disambiguated correctly, obtaining an accuracy of 60.65%. Since there is no system developing this task in Spanish, we cannot compare our results to the results obtained with other approaches. Therefore, we create a baseline, following the methodology of the state of the art [13], [18]: the baseline offers the most frequent relation showed by the marker (in our case, in the training corpus of the RST Spanish Treebank). The baseline obtains an accuracy of 49.18%. For this reason, we consider that the results obtained in our experiments are representative, and constitute the first attempt towards the disambiguation of discourse markers senses in Spanish.

After a qualitative evaluation, we observe that the rules including lists of semantically related verbs (basically, the rules included in Tables 9 and 10) are useful, but they would have better performance if they included more verbs. In this study, we only include in the lists (@result, @interpretation and @means) the verbs found in the training corpus. In the test corpus, some contexts including different but semantically related verbs are detected. For example, *originar* (“to origin”), *traer* (“to bring along”) and *tener como consecuencia* (“to have as a consequence”) are semantically related to the verbs of the list @result; and the verbs *partir* (“to start from”), *iniciar* (“to begin”) and *abordar* (“to deal with”) are related to the verbs of the list @means. In addition, for some markers (such as *pues* and *ya que*), a few contexts are retrieved from the test corpus, so it is difficult to assess the performance of the corresponding rules.

6 Conclusions

In this paper, a symbolic approach to detect and solve the ambiguity of discourse markers in Spanish texts is presented. Specifically, we deal with discourse sense ambiguity, i.e. with markers that can signal more than one rhetorical relation (in this work, RST relations). The proposal is mainly based on syntactic and lexical features, and not on punctuation, as it has been done until now for Spanish. The performance of the approach is better than the baseline created following the methodology of the state of the art.

Although the results are encouraging, we are conscious that there is room for improvement. Specifically, as a future work, we will evaluate each rule individually, not the approach as a whole. Regarding the lack of contexts for the evaluation of some rules, [25] states that there are two possible strategies: a) to leave the corpus as it is, with few or no examples of some cases (but the problem will be the lack of training examples for machine learning systems), or b) to add low-frequency examples artificially in order to "enrich" the corpus (but the problem will be the distortion of the native frequency distribution and perhaps the confusion of machine learning systems). In the future, we plan to follow the second option, that is, to compile a specific corpus including contexts with ambiguous discourse markers, annotate it manually and then re-evaluate the problematic rules.

In addition, we plan to integrate semantic verbal information in the rules, to solve the problem detected in the qualitative evaluation, as mentioned in section 5. We will use lexical databases, such as EuroWordNet (http://www.illc.uva.nl/EuroWordNet).

Finally, in the future, our disambiguation approach will be integrated in a discourse parser for Spanish, and several related applications will be developed (automatic summarization and information extraction, among others). Also, we would like to combine our symbolic approach with machine learning methods, in order to examine the performance of a hybrid disambiguation system.

References

1. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243-281 (1988)
2. Marcu, D.: The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. Computational Linguistics 26(3), 395-448 (2000)
3. Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. In: Proceedings of the 2003 Conference of NAACL-HLT, pp. 149-156 (2003)
4. Subba, R., Di Eugenio, B.: An effective discourse parser that uses rich linguistic information. In: Proceedings of the 2009 Conference of HLT-ACL, pp. 566-574 (2009)
5. Sumita, K., Ono, K., Chino, T., Ukita, T., Amano, S.: A discourse structure analyzer for Japanese text. In: Proceedings of the International Conference on Fifth Generation Computer Systems, pp.1133-1140 (1992)
6. Pardo, T.A.S., Nunes, M.G.V.: On the Development and Evaluation of a Brazilian Portuguese Discourse Parser. Journal of Theoretical and Applied Computing 15(2), 43-64 (2008)
7. da Cunha, I., San Juan, E., Torres-Moreno, J-M., Cabré, M.T., Sierra, G.: A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra-

sentence Discourse Segments in Spanish. In: Gelbukh, A. (ed). Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science (LNCS). CICLing 2012, Part I, 7181, pp. 462-474 (2012)
8. Maziero, E., Pardo, T.A.S., da Cunha, I., Torres-Moreno, J-M., SanJuan, E.: DiZer 2.0 - An Adaptable On-line Discourse Parser. In: Proceedings of the III RST Meeting (8 th Brazilian Symposium in Information and Human Language Technology) (2011)
9. Taboada, T.: Discourse markers as signals (or not) of rhetorical relations. Journal of Pragmatics 38, 567-592 (2006)
10. Pitler, E., Nenkova, A.: Using Syntax to Disambiguate Explicit Discourse Connectives in Text. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 13-16 (2009)
11. van Dijk, T.A.: Texto y contexto (Semántica y pragmática del discurso). Madrid: Cátedra (1984)
12. Hirschberg, J., Litman, D.J.: Empirical studies on the disambiguation of cue phrases. Computational Linguistics 19(3), 501–530 (1993)
13. Miltsakaki, E.; Dinesh, N.; Prasad, R.; Joshi, A.; Webber, B.: Experiments on sense annotations and sense disambiguation of discourse connectives. In Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT2005) (2005)
14. Miltsakaki, E., Prasad, R., Joshi, A., Webber, B.:. The Penn Discourse Treebank. In 4th International Conference on Language Rescourses and Evaluation (LREC 2004), 2004.
15. Sporleder, C., Lascarides, A.: Exploiting Linguistic Cues to Classify Rhetorical Relations. In Proceedings of Recent Advances in Natural Language Processing (2005)
16. Bayerl, P.S.: Disambiguierung deutschsprachiger Diskursmarker: Eine Pilot-Studie. Linguistik Online, 18 (2004)
17. Versley, Y.: Multilabel Tagging of Discourse Relations in Ambiguous Temporal Connectives. In: Proceedings de la 8th International Conference on Recent Advances in Natural Language Processing, pp.154-161 (2011)
18. Al-saif, A., Markert, K.: Modelling Discourse Relations for Arabic. In: 2011 Conference on Empirical Methods in Natural Language Processing (2011)
19. Prada, J.J.: Marcadores del discurso en español. Análisis y representación. Master Thesis. Uruguay: Facultad de Ingeniería, Universidad de la República (2001)
20. Koza, W.A.: Detección automática de marcadores discursivos del español una aplicación con xfst. Philologica Urcitana. Revista de iniciación a la investigación en Filología 7, 59-74 (2012)
21. da Cunha, I., Torres-Moreno, J-M., Sierra, G.: On the development of the RST Spanish Treebank. In: Proceedings of the Fifth Law Workshop (ACL 2011), pp. 1-10 (2011)
22. da Cunha, I., Torres-Moreno, J-M., Sierra, G., Cabrera-Diego, L-A., Castro-Rolón, B-G.; Juan-Miguel Rolland-Bartilotti The RST Spanish Treebank On-line Interface. In: Angelova, G. et al. (eds.). *Proceedings of Recent Advances in Natural Language Processing*. pp. 698-703 (2011)
23. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: FreeLing 1.3. Syntactic and semantic services in an open-source NLP library. In: N. Calzolari et al. (ed.). Proceedings of the Conference LREC 2006, pp. 48-55 (2006)
24. da Cunha, I., SanJuan, E., Torres-Moreno, J-M., Lloberes, M., Castellón, I.: DiSeg 1.0: The First System for Spanish Discourse Segmentation. Expert Systems with Applications 39(2), 1671-1678 (2012)
25. Hovy, E.: Annotation. A Tutorial. Presented at the 48th Annual Meeting of the Association for Computational Linguistics (2010)

Parsing Intentions of Speech

Ke Wang[1], Gerald Penn[2], Rongpei Wang[3]

[1, 3] Dalian University of Technology, 116024 Dalian, China
[2]University of Toronto, 10 King's College Road, M5S 3G4, Canada
[1]wang.coco.ke@gmail.com, [2]gpenn@cs.toronto.edu, [3]rpwang1@hotmail.com

Abstract. This paper is aimed at revealing the inner structures of intentions of speech through demonstrating a method of parsing intentions of speech. The intentions of speech are parsed separately, which is inspired by Searle's notation $F(p)$. In this research, both the propositional content and the illocutionary forces are represented as semantic frames, the categories of which are combined together with syntactic rules of Categorial Grammar. This method is implemented in a grammar system programed in ALE a logic programming language based on the logic of Typed Feature Structures.

1 Introduction

The studies of intentions are aimed at designing intelligent agents which can understand human's intentions through communication. In the 80s and 90s of last century, many scientists coming from many fields began to research in this area. And some results of those researches have been applied to very practical usages. James Allen has developed his ideas in his early study of intention [1] into a language processing model TRIPS [2] which is used as a very important section in the intelligent task learning system PLOW [3]. Barbara Grosz and Candace Sidner developed their theory of shared plan [10] into a model of collaboration COLLAGEN [11] which is later used as the most critical part of a series of robotic systems [12]. Philip R. Cohen and Hector Levesque [7, 8] considered the intention of a conversation as a persistent goal over the whole joint activity, which is formalized as a true proposition in some possible world. Later, Levesque developed his idea about possible worlds into a logic programing language GOLOG [14].

However, there is still one problem left for the researchers to conquer; that is the inner structures of intentions of speech. In Allen et al.'s paper [2], though they "need to produce a detailed semantic representation of what was said-something that captures what the user meant by the utterance"; however, at that time, "the only way to get such system is to build it by hand". They couldn't make it because there has not yet been available a complete theory of the role of intentions in communication: how smaller intentions combine to form composite ones.

This paper is an attempt to reveal the syntactic-semantic structure of intentions of speech, and hence how the small intentions of an utterance are composed as a whole. This research is inspired by Searle's notation $F(p)$, i.e. the illocutionary force and propositional content are parsed separately. Both the illocutionary force and propositional content are represented as semantic frames; and the combination rules

are borrowed from Categorial Grammar. In order to implement our ideas, we constructed a grammar system which is programmed in ALE, a logic programming language based on the logic of typed feature structure. We will introduce it in detail in section 3. And in section 4, we will discuss about how to broaden the coverage of the utterances with probability-based methods.

2 Methods used in this paper

The main idea of the study of intentions of speech can be traced back to J. L. Austin's theory about speech act [4]. He describes speech act as three levels, i.e., locution, illocution, and perlocution. For example, a speaker said "it's dark" meaning that the light is not bright enough. Austin calls this kind of "saying something" locutionary act. By saying "it's dark", the speaker might be conveying his view about the situation, and may indirectly convince the hearer to turn on the light. This is called illocutionary act by Austin. The result of "saying so" is that the hearer turned on the light; and this is exactly what the speaker wants him to do. The final immediate result of "saying so" is named perlocutionary act. John R. Searle employs the term "illocution", but he does not accept Austin's distinction. He [17] argues that there is a distinction between illocutionary force and propositional content of the illocutionary act, as shown in the following sentences:

(1) Sam smokes habitually.

(2) Does Sam smoke habitually?

(3) Sam, smoke habitually!

(4) Would that Sam smoked habitually.

The first sentence is an assertion; the second one is a question; the third is an order; and the fourth a wish or desire. They are of different illocutionary acts, though they share the same propositional content. Searle uses $F(p)$ to represent the relationship between illocutionary force and propositional content, where F represents the illocutionary force, and P represents the propositional content. In this paper, we adopt Searle's notation; and we assume intention of speech could be rewritten as $F(p)$.

Searle's study opens up a possibility for us to look into the detail structure of intentions of speech. Searle's notation $F(p)$ inspired us, that we could parse the intentions of speech by recognizing the illocutionary force and the propositional content separately. In this paper the propositional content is represented by Semantic Frame [9], for usually propositions could be roughly described as a relationship held by the predicate and its arguments. For example, in (5), <bring> is the predicate; and "you", "me", and "book" are the arguments.

(5) Could [[you]$^{\text{GIVER}}$ <bring> [the book]$^{\text{THEME}}$ [to me]$^{\text{RECEIVER}}$]$^{\text{PROPOSITION}}$?

Searle also suggests to "…distinguish two elements in the syntactical structure of the sentence, which we might call the propositional indicator and the illocutionary force indicator." And he views the illocutionary indicators as a kind of devices, by which the illocutionary act is performed. In modern English, those indicators includes at least: "word order, stress, intonation contour, punctuation, the mood of the verb", and so on. For example, (5) is a question. We recognize it as a question because the auxiliary "could" was moved to the front of the utterance and because we saw a question mark at the end of the utterance. In other words, those illocutionary force indicators activated a question frame in our minds, thus we could recognize it as a question. For example, this question could then be rewritten into (6):

(6) [[Could]$^{\text{AUXILIARY}}$ [you bring the book to me]$^{\text{PROPOSITION}}$] [?]$^{\text{Q-MARKER}}$] $^{\text{QUESTION}}$

However, Frame Semantics does not provide any composition rules. In this paper, we take advantage of the syntactic rules of Categorial Grammar [13, 20] to combine the categories. For instance, the semantic structure of the proposition of (5) could be obtained by the way of (7):

(7)

Could	you	bring	the book	to me	?
	GIVER	<bring>\GIVER/RECEIVER/THEME	THEME	RECEIVER	
		<bring>\GIVER/ RECEIVER			
		<bring>\GIVER			
		<bring>			

The semantic structure of illocutionary force of (6) could be obtained by the way of (8), where "[… …]" represents the very abstract question frame (PROP=PROPOSITION). We say that the auxiliary "could" evokes the question frame, or in other words the question frame is assumed when we meet "could"; and then the rest of the frame are tested through the following steps of parsing.

(8)

Could	[… …]	you bring the book to me	?
AUX	query/Q-MARKER/PROP\AUX	PROPOSITION	Q-MARKER
	query/Q-MARKER/PROP		
	query/Q-MARKER		
	query		

Note that PROPOSITION in (8) should be further structured as the frame labeled with <bring> in (7). They are combined together by an operation called "node-unification"; refer to the details in section 3.3.

3 Grammar System

The grammar system includes a type hierarchy and a set of grammar rules. The type hierarchy composes of a taxonomy of intentions of speech (refer to 3.1), and some

linguistic knowledge, such as classification of verbs (over 600 English verbs), definitions of illocutionary force indicators, definitions of semantic roles (over 140 semantic roles), and so on. In the type hierarchy, each type is in accord with a set of features. The classification of verbs is aimed at assigning appropriate arguments to the predicate through treating the arguments as the features of verbs. The grammar rule is an integration of Frame Semantics and Categorial Grammar (see Section 2). This grammar system is only fit for parsing simple sentences, i.e., compounded sentences, like if-then-construction, are not covered.

3.1 Taxonomy of Intentions of Speech

Searle classified illocutionary acts into five types, i.e., assertive, directive, expressive, commissive, and declaration. According to Searle [18, p12-15], an assertive is to "commit the speaker … … to the truth of the expressed proposition", like (1) above; directive is "to get the hearer to do something", such as (3); expressive is "to express the psychological state … … specified in the propositional content", such as (4); declaration is about how a "successful performance (of a speech act, noted by authors) guarantees that the propositional content corresponds to the world", like "you are fired"; and commissive is to "commit the speaker to some future course of action", for instance, "I will come". Apparently, Searle's classification is not specific enough if we want to use it for parsing. Besides, they are not concerned with the linguistic features of utterances, such as, for which features that an utterance is to be recognized as a query, and for which features an utterance is to be identified as an imperative. In this research, we classified intentions of speech into more specific types keeping both their linguistic manifestations and their philosophical significance into perspective, such as Searle's philosophy about social construction, see Fig. 1.

The reasons for that the intentions of speech are classified into "to describe the world", "query", and "imperative" are simple. We are driven to speak, because we want to transmit information, or we want to ask for information, or we want to give an order. Roughly, "to describe the world" corresponds to assertive; and "query" and "imperative" are equal to Searle's directive. But we have some disagreements with Searle on the classification of assertive, expressive, and commissive, and on the explanation of declaration force. We think both expressive and commissive are assertive, when we look at the inner world from outside. If assertive is "to commit the speaker to something's being the case, to the truth of the expressed proposition", then commissive is "to commit the speaker to some future course of action" by truthfully stating the inner world of the speaker. For example, when a speaker said "I will come", he must already have had a plan in his mind. Similarly, an expressive is to "express the psychological state" of the speaker. Searle didn't think of them as assertive, probably because that it is hard to detect the truth condition of people's inner world. But we assume that all of these happen under the truthfulness conditions, so it is possible and feasible for an observer to detect people's inner world. This is one of the reasons that we separate "third personal behavior" from "brute facts". The other disagreement is about the force of declaration. Searle thinks the truth condition of this kind of illocutionary act depends on some institutions, so he calls the facts caused by declarations, such as nominating, "institutional fact". Institutional fact is also called

social fact by Searle in [19]. However, the truth condition of many facts does not obviously depend on the power of an institution, such as "snow is white". The reason that W-H-I-T-E represents the meaning white is that it is recognized by the social members who use it as a symbol to describe colors. Or in other words it is comprehensible because the social members reached an agreement that this word represents the color of snow. So, we define the intentions of this kind of utterances as to describe "social fact". The discussion of linguistic philosophy may need a book-length, so we just stop at here. We hope that we could have another opportunity to discuss it in a later paper.

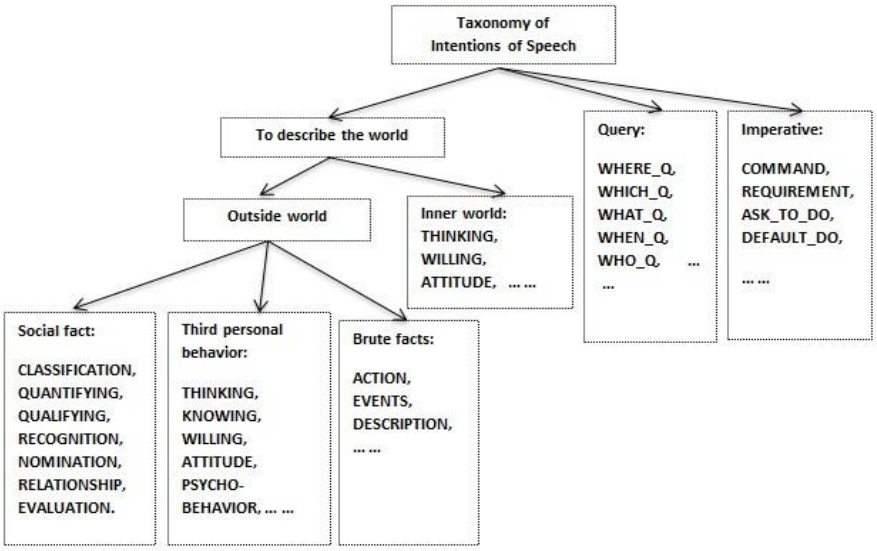


Fig. 1: Taxonomy of Intentions of Speech.

Table 1 shows some syntactic features of utterances, according to which intentions of speech are classified into types. As it is shown, each type of utterances is described as a distinctive chunk labeled with a semantic role, such as CLASS in "classification", and a set of syntactic elements, such as "be", "action_verb", "?", and so on. Some of the distinctive chunks are realized by a specific set of words, such as ATTRIBUTE is realized by attribute adjectives. Some distinctive chunks are realized by specific sets of intentions of speech. For example, CLUE in "recognition" is realized by the implementation of "brute fact", and the WILL in "willing" is realized by the implementation of "future fact". This is the way that some complex intentions of speech are composed; and this is also the way how the grammar rules are compiled, such as (5) through (8).

Table 1: Syntactic features of utterances.

Intentions & Distinctive chunks	Linguistic features	Examples

classification	TARGET + be + CLASS	Apple is a kind of fruit.
CLASS	class words	fruit, teacher, thief, … …
quantifying	NUMBER + TARGET	ten dollars
qualifying	ATTRIBUTE + TARGET	red shirt
ATTRIBUTE	attribute adjectives	red, tall, good, bad, … …
recognition	DEMONSTRATIVE + be + TARGET + CLUE	This is the book I bought yesterday
CLUE	brute_fact	I bought yesterday
nomination	TARGET/DEMONSTRATIVE + NAME	This/He is Jack.
NAME	proper names	Jack, Harvard University
relationship	ITEM_1 + relation_verb/be + ITEM_2	Three plus three is six.
evaluation	TARGET + ATTRIBUTE + than + STANDARD	He is taller than me.
	TARGET + ATTRIBUTE+CLASS + SCOPE	He is the tallest student in his class.
brute_fact	PROPOSITION (past and present tense)	Here comes the bus.
PROPOSITION	frame	
3rd_ thinking	Third person+think/… … + THINK	He thought this is correct/I like it.
THINK	social_fact/third_personal_behavior	this is correct/I like it
3rd_knowing	Third person+know/… …+KNOWN	He knows that guy is Jack/she loves chocolate/she bought the book… …
KNOWN	"to describe the world"	That guy is jack/she loves chocolate/she bought the book
willing	COGNIZER + promise/wish… … + WILL	I wish he will come to the party.
WILL	future_fact	He will come to the party
Wh_query	Wh_pronoun + QUERY + ?	When could you bring me the book?
QUERY	query	
query	auxiliary + PROPOSITION + ?	Could you bring me the book?
imperative	PROPOSITION (action_verb in present tense)	Listen! Let's go!

3.2 Parsing

The grammar system is implemented in ALE [5], a logic programming language based on the logic of typed feature structure [6, 15]. A typed feature structure is a directed graph possibly with cycles. The nodes on the graph are labeled with types; and the edges between nodes are labeled with features, see Fig. 2 A typed feature structure could be represented either by a graph, like (a), or by an AVM, like (b).

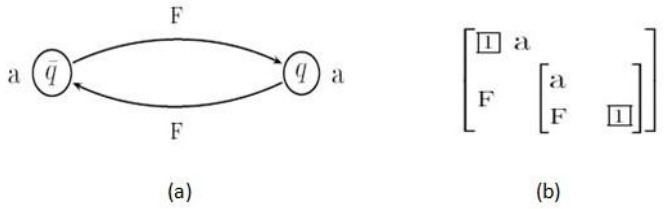


Fig. 2: The graphic and AVM representation of feature structure

We chose to describe semantic frame with typed feature structure; for one reason, it is more similar to the concept of frame, the other reason is that dependence-tee and phrase-structure-tree are not suitable for representing intentions of speech, as shown in Fig. 3.

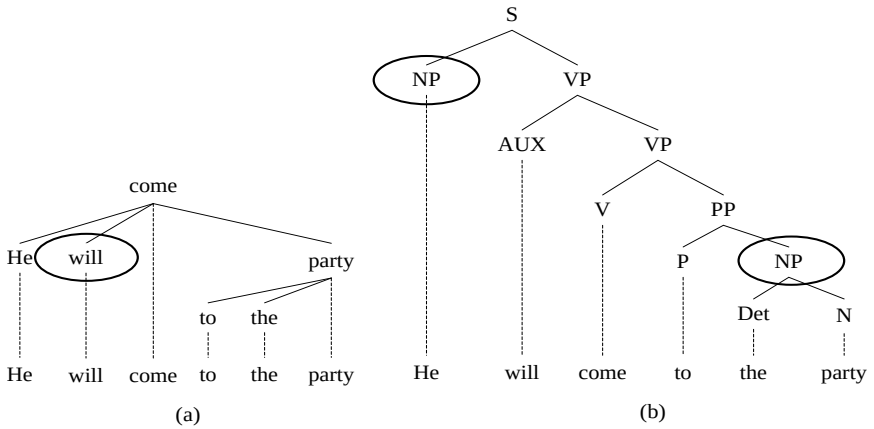


Fig. 3: The dependence-tree and phrase-structure tree of "he will come to the party".

In Fig. 3 above, (a) and (b) are the dependence tree and the phrase-structure tree respectively. Apparently, in (a) "will" is not argument of "come", though it syntactically depends on it. In (b), the two NPs are not on the same level, but semantically, they are equally treated as the arguments of the predicate. If either dependence-tree or phrase-structure tree were employed, the parsing task would be made too complicated to be accomplished.

The only one operation in the logic of typed feature structure is Unification. This is very similar to the unification operation in PROLOG except that it is constrained by "type consistency". For example, in (9), whether the two feature structures are unifiable depends on if "b" and "c" are consistent. Carpenter [6, p12] used to give a definition of consistency. According to him, they are unifiable, only when b is a subtype or supertype of c.

(9) $\begin{bmatrix} a \\ \text{F}: [b] \end{bmatrix} \sqcup \begin{bmatrix} a \\ \text{F}: [c] \end{bmatrix} = \text{true or false?}$

In order to parse the semantic structures of intentions of speech, we need to translate the application rules of Categorial Grammar into unification operation. We replace "/" and "\"in categorial grammar with "⊔" the unification operation in the logic of typed feature structure. For example, the semantic structure of (8) could be obtained through a series of operations as the followings1. The order of the feature structures on the left side of "=" indicates the parsing order.

(10) $\begin{bmatrix} giver \\ GIVER\_E: [giver\_element] \end{bmatrix} \sqcup [you] = \begin{bmatrix} giver \\ GIVER\_E: [you] \end{bmatrix}$

(11) $\begin{bmatrix} receiver \\ RECEIVER\_E: [receiver\_element] \\ SYN: [receiver\_particle] \end{bmatrix} \sqcup [to] \sqcup [me] = \begin{bmatrix} receiver \\ RECEIVER\_E: [me] \\ SYN: [to] \end{bmatrix}$

(12) $\begin{bmatrix} theme \\ THEME\_E: [theme\_element] \end{bmatrix} \sqcup [the\ book] = \begin{bmatrix} theme \\ THEME\_E: [the\ book] \end{bmatrix}$

(13) $\begin{bmatrix} giver \\ GIVER_E: [you] \end{bmatrix} \sqcup \begin{bmatrix} < bring > \\ GIVER: [giver] \\ RECEIVER: [receiver] \\ THEME: [theme] \end{bmatrix} \sqcup \begin{bmatrix} theme \\ THEME_E: [the\ book] \end{bmatrix} \sqcup$

$$\begin{bmatrix} receiver \\ RECEIVER\_E: [me] \\ SYN: [to] \end{bmatrix} = \begin{bmatrix} < bring > \\ GIVER: \begin{bmatrix} giver \\ GIVER\_E: [you] \end{bmatrix} \\ RECEIVER: \begin{bmatrix} receiver \\ RECEIVER\_E: [me] \\ SYN: [to] \end{bmatrix} \\ THEME: \begin{bmatrix} theme \\ THEME\_E: [the\ book] \end{bmatrix} \end{bmatrix}$$

(14) $[could] \sqcup \begin{bmatrix} query \\ AUX: [auxiliary] \\ PROP: [frame] \\ MARKER: [q\_marker] \end{bmatrix} \sqcup \begin{bmatrix} < bring > \\ GIVER: \begin{bmatrix} giver \\ GIVER\_E: [you] \end{bmatrix} \\ RECEIVER: \begin{bmatrix} receiver \\ RECEIVER\_E: [me] \\ SYN: [to] \end{bmatrix} \\ THEME: \begin{bmatrix} theme \\ THEME\_E: [the\ book] \end{bmatrix} \end{bmatrix} \sqcup [?] =$

[1] Considering the convenience of reading and the length of this paper, because the actual semantic structure of an utterance could be very complex, we only show some simple examples as the illustration of parsing.

$$\begin{bmatrix} query \\ AUX: [could] \\ PROP: \begin{bmatrix} < bring > \\ GIVER: \begin{bmatrix} giver \\ GIVER\_E: [you] \end{bmatrix} \\ RECEIVER: \begin{bmatrix} receiver \\ RECEIVER_E: [me] \\ SYN: [to] \end{bmatrix} \\ THEME: \begin{bmatrix} theme \\ THEME\_E: [the\ book] \end{bmatrix} \end{bmatrix} \\ MARKER: [?] \end{bmatrix}$$

(10) through (12) are to label each category with a semantic role. It is assumed that what semantic role a category is to have depends on the category's lexical meaning and on the category's syntactic environment. In (10), as long as "you" is predefined as a subtype of "giver_element", it is unifiable with the "node" $[giver\_element]$. In (11), "me" is labeled with "receiver", for it is preceded by particle "to". Usually, in similar constructions, "me" is replaceable, such as "bring it to Jack", "bring it to him", etc., but "to" is relatively irreplaceable. So, it is reasonable to say that "to" is functioned as a kind of "semantic role indicator" (refer to Appendix). (13) is to parse the propositional content. Each argument is fulfilled with a typed category by the way of "node-unification". (14) is intended for parsing the intentions of speech, i.e., the propositional content and other syntactic elements are to be merged with the nodes of the predefined illocutionary force frame. "Could", "?", these illocutionary force indicators that are considered as the features of "query", are planted at "AUX" and "MARKER" respectively; and the proposition <bring> is unified with the node $[frame]$.

4 Discussion and Future Works

In this paper, we presented an idea and a method of parsing intentions of speech. The idea to parse the illocutionary force and the propositional content separately is inspired by Searle's notation $F(P)$. In order to implement this idea, we developed a grammar system. Theoretically, it is very hard to test the coverage of the grammar system. It at present includes about 50 rules, more than 600 verbs, and hundreds of nouns; however, this doesn't mean it could only process 50 utterances. The limitation of the grammar system is that each utterance to be parsed has to correspond to a set of grammar rules. But, we believe it is hopeful to develop methods to broaden the coverage of this grammar system. For example, since it is assumed that each language with infinite utterances could be described by a set of finite grammar rules; therefore, it is possible and reasonable to cut a compounded sentence into clauses and then send these clauses to the grammar system. With these parsed clauses, we could, with manual labor, obtain a probability table which could be about how possible it is to predict two clauses to be joined together. Besides, we are planning to construct an intelligent agent to test our hypothesis, i.e. finer semantic representation will promote the quality of understanding intentions of humans.

References

1. J. F. Allen, and C. R. Perrault, Analyzing intention in Utterances. Artificial Intelligence, Vol. 15, p143-178 (1980)
2. J. F. Allen et al., Towards Conversational Human-computer Interaction, AI Magazine (2001)
3. J. F. Allen et al., PLOW: A Collaborative Task Learning Agent. National Conference on Artificial Intelligence, Vancouver, BC (2007)
4. J. L. Austin, How to Do Things with Words, Cambridge University Press (1962)
5. B. Carpenter, and G. Penn, The Attribute Logic Engine: User's Guide. This is downloadable from www.cs.toronto.edu/~gpenn/ale.html
6. B. Carpenter, the Logic of Typed Feature Structures, Cambridge University Press (1992)
7. P. Cohen, and H. Levesque, Persistence, Intention, and Commitment in Intentions in Communication edited by Philip R. Cohen et al., the MIT Press (1990)
8. P. Cohen, and H. Levesque, Rational Interaction as the Basis for Communication in Intentions in Communication edited by Philip R. Cohen et al, the MIT Press (1990)
9. C. Fillmore, The Case for Case in Bach and Harms (edit): Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, page 1-881 (968)
10. B. Grosz, and C. Sidner, Plans for Discourse in Intentions in Communication edited by Cohen, Philip R. et al. The MIT Press, London, England (1990)
11. C. Rich, et al, COLLAGEN Applying Collaborative Discourse Theory to Human-computer Interaction. AI Magazine, Vol. 22, No. 4 (2001)
12. C. Rich, and C. Sidner, Robots and Avatars as Hosts, Advisors, Companions, and Jesters, AAAI Spring symposium (2009)
13. J. Lambek, The Mathematics of Sentence Structure, American Math Monthly, Vol. 65, No. 3 (1958)
14. H. Levesque, et al., GOLOG: A Logic Programming Language for dynamic domains, Journal of Logic Programming, Vol. 31, p59-84 (1997)
15. G. Penn, The Algebraic Structure of Attributed type Signatures, doctoral thesis of Carnegie Mellon University (2000)
16. R. Perrault, An Application of Default Logic to Speech Act Theory in Intentions in Communication edited by Philip R. Cohen et al, the MIT Press (1990)
17. J. R. Searle, Speech Acts: An Essay in the philosophy of Language. Cambridge University Press (1969)
18. J. R. Searle, Expression and Meaning: Studies in the Theory of Speech Acts. Cambridge University Press (1979)
19. J. R. Searle, the Construction of Social Reality. Allen Lane the Penguin Press (1995)
20. M. M. Wood, Categorial Grammars, Routledge, London and New York (1993)

Appendix:

Table 2: Semantic role indicators

Semantic roles	Semantic role indicators	Semantic roles	Semantic role indicators
PLACE	at, above, after, around, beyond, by,	GROUND	from

	in, inside, on, outside, within, round	ATITTUDE	against
SCOPE	near, under, up, upon, before,	SUPPORT	against, on
	behind, below, between, over, to	REFERENCE	than, like, against
POSITION	across, among, between, beyond,	AGENT	by
	in, within, under	COGNIZER	to
PATH	out, back	REASON	since, by
TIME	at, after, ago, around, before,	MEANS	by, through
	between, beyond, by, during, in,	COMPANY	with, and
	inside, on, outside, over, within, for,	CAUSE	at, by, with, from, through, under
	near, past, round, since, through, till, until		
DIRECTION	at, in, to, for, towards, into	TOOLS	with
PURPOSE	at, for, to	WAY	at, by, in, on
ASPECT	at	MANNER	with, in,
CAUSER	with, for, about	TOPIC	on, over, about
ORIGIN	from	DISTANCE	for
ADDRESSEE	to	INGREDIENT	from
NAME	under	RECEIVER	to

Feature Analysis for Paraphrase Recognition and Textual Entailment

Andrea Segura-Olivares, Alejandro García, and Hiram Calvo

Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN), Mexico City, Mexico
msegura_b12@sagitario.cic.ipn.mx,
igarcia_b12@sagitario.cic.ipn.mx, hcalvo@cic.ipn.mx

Abstract. Paraphrase recognition is the task of Natural Language Processing of detecting if an expression restated as another expression contains the same information. Textual Entailment recognition, while being similar to paraphrase recognition, is a task that consists in finding out if a given text can be observed as a consequence of another text fragment, sometimes considering only part of the original meaning, or adding some inferences based on common sense. Traditionally, for solving this problem, several *lexical, syntactic* and *semantic* based techniques are used. In this work, we seek to use the less resources as possible, while being effective. For this, we perform a feature analysis for performing Paraphrase Recognition and recognizing Textual Entailment experimenting with the combination of several Natural Language Processing techniques like *word overlapping, syntactic analysis*, and *elimination of stop words*. Particularly, we explore using the syntactic n-grams technique combined with some auxiliary approaches such as *stemming, synonym detection, similarity measures* and *linear interpolation*. We measure and compare the performance of our system by using the Microsoft Research Paraphrase Corpus, and the RTE-3 test set for Paraphrasing and Textual Entailment, respectively. Syntactic n-grams produce good results for Paraphrase Recognition. As far as we know, syntactic n-grams had not been used for this task. For Textual Entailment, our best results were obtained by using a simple word overlapping algorithm based on stemming and elimination of stop words.

1 Introduction

The study of new techniques in Natural Language Processing (NLP) has become more popular recently between researchers. As a consequence, there are several tasks that are considered solved on this important area; however, NLP still has some challenges that need to be solved. Some of them are Paraphrase Recognition and Textual Entailment.

The best results reported for these tasks usually have one characteristic in common, and it is that they require costly resources such as combinations of lexical analysis, semantics, logic inference, background knowledge and machine learning [10].

For languages like English, having the required resources may not be a problem because English has been studied for several people, and there are many NLP tools

⋆ Work done under support of CONACyT-SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN

that can be used for the purpose of this kind of systems. However, for other languages that have not been studied deeply, resources could be a restriction for implementing NLP systems. Therefore, the purpose of our work was to create a system that employed as less resources as possible but still having a good performance on the Paraphrase Recognition and Textual Entailment tasks.

In the following sections we will describe two standard tasks for Paraphrase Recognition, and Textual Entailment, as well as the general methods for tackling them. For this work we experimented with several NLP Techniques (See Section 2), and particularly we propose using syntactic n-grams, described in Section 2.3. We present our experiments and their evaluation with two respective gold standards in Section 3 for Paraphrase Recognition, and in Section 4 for Textual Entailment. Finally we draw our conclusions in Section 5.

Paraphrase Recognition Paraphrasing is the process of restating an expression E_1 in other expression(s) E_2, E_3, ..., E_n that convey the same meaning. For instance, the expression:

- $\mathbf{S}_1$: Juan Rulfo wrote "El llano en llamas".
- $\mathbf{S}_2$: "El llano en llamas" was written by Juan Rulfo.

are paraphrases of each other.

In general the paraphrase processing is divided in three main tasks that are described below:

1. **Extraction**: this task has the goal of obtaining a set as large as possible of pairs (S_1, S_2) that conform a paraphrase pair from a big corpus given as input of the system.
2. **Generation**: this task has the objective to yield a set of expressions (S_2, S_3, ..., S_n) as large as possible that are paraphrases of the input string S_1.
3. **Recognition**: this task has as objective to detect if two given expressions (S_1, S_2) given are paraphrases of each other.

Textual Entailment Textual Entailment recognition is the task of finding out whether the semantics of a text can be inferred from the semantics of another text. The entailing and entailed text are termed *text (T)* and *hypothesis (H)* respectively. An example of textual entailment is the following:

- T: The drugs that slow down Alzheimer's disease work best the earlier you administer them.
- H: Alzheimer's disease can be slowed down using drugs.

where we can see that H can be inferred from T. Therefore, this is a true textual entailment pair.
An example of a non textual entailment pair is the following:

- T: It is important to stress that this is not a confirmed case of rabies.
- H: A case of rabies was confirmed.

where the semantics of the text H cannot be inferred from the text T.

Differently to Paraphrase Recognition, Textual Entailment is a directional relation because the hypothesis can be inferred from the text but not necessarily the opposite way.

Many NLP tasks like document summarization (SUM),Information Retrieval (IR), Information Extraction (IE) and Question Answering (QA) can take advantage of Paraphrasing and Textual Entailment.

The most common approaches for these tasks and their required resources are:

- **Logic based**: theorem provers, knowledge bases, inference rules and logical conversions.
- **Machine learning based**: annotated examples.
- **Decoding based**: substitution rules, knowledge bases.
- **Semantics based**: semantic networks.
- **Syntactic based**: syntactic parsers.

In this work we experiment with the impact of lexical, syntactic and semantic techniques for Paraphrase Recognition and Textual Entailment.

The *lexical approaches* operate directly with the input strings without making important changes to them; however, sometimes pre-processing is required.

The *syntactic approaches* aim to analyze sentences to show how their words interact with each other. For this approach we need to obtain a syntactic tree that can be obtained by using a syntactic parser.

The *semantic approaches* usually operate on a shallow semantic level; more specifically, for this work we use *Lexical Semantics* which mainly consist of similarity measures between words and semantic relations (hypernym, hyponym, meronymy, holonymy, ..., etc.) obtained from thesauri or semantic networks like WordNet.

We experiment with several combinations of these approaches and auxiliary techniques to find out which combination has the best performance. In the following sections we describe the implemented NLP techniques, and report the results of each experiment that we performed.

2 Implemented NLP techniques

In this section we present the lexical, syntactic and semantic techniques we implemented for Paraphrase Recognition and Textual Entailment.

Particularly we will show examples for Textual Entailment, being **T** the Text and **H** the Hypothesis, although many of these techniques were used for Paraphrasing as well, considering indistinctly one expression of the paraphrasing as Text, and other as Hypothesis. See Section 3 for details on the experiments for Paraphrase Recognition.

2.1 Lexical Module

In this section we describe the Lexical Module we implemented. This module operates at a shallow level of the given texts.

The general algorithm for the lexical module consists in measuring the ratio of coverage of the hypothesis by the given text, this means that, the more words are covered in the hypothesis, the more likely they are to be a textual entailment pair.

We use a coverage threshold **TH** to decide if a given pair is or is not a textual entailment pair; that is, if the coverage ratio of **H** is greater or equal than **TH** the answer will be "YES" otherwise the answer is "NO".

For example, given the pair:

- **T**: The Aztecs were a civilization based on war. Most of them were warriors.
- **H**: The Aztecs were warriors.

we can see that the words of **H** that appear on **T** are "The," "Aztecs," "were" and "warriors." Therefore, the coverage ratio of **H** is $\frac{4}{4} = 1$ so, if **TH** were 0.7 then the answer would be "YES".

The general algorithm for this process is the following:

Given a pair of expressions T, H and a threshold TH:

$H \leftarrow$ preprocessing(H)
$T \leftarrow$ preprocessing(T)
$L_H \leftarrow$ length(H)
$common \leftarrow 0$
for all words *w* in H **do**
  if $contains(T, w)$ **then**
    $common \leftarrow common + 1$
  end if
end for
$coverage \leftarrow common/L_H$
if $coverage \geq TH$ **then**
  return "YES"
else
  return "NO"
end if

The main part of the algorithm is the preprocessing step which receives a text fragment (T or H) and changes it with an auxiliary technique like *stemming, stop words, similarity measures* or *negation detection* in order to obtain a better result. The following sections describe how these techniques are applied.

Preprocessing: Removing Stop Words. Like we mentioned before, stop words are words that in many cases can be removed from a natural language text fragment without loosing critical information, because stop words are very frequent words that appear in most of the text fragments, and therefore the information conveyed by these words usually is not relevant. There is not a definitive list of stop words; however, most common stop word lists consists of *prepositions* and *determiners*.

Our preprocessing system that handles stop words takes as input a text fragment and returns it without the stop words that it contains. For example:

Input: Bountiful arrived after war's end, sailing into San Francisco Bay 21 August 1945. Bountiful was then assigned as hospital ship at Yokosuka, Japan, departing San Francisco 1 November 1945.
Output: Bountiful arrived war's end, sailing San Francisco Bay 21 August 1945. Bountiful was assigned hospital ship Yokosuka, Japan, departing San Francisco 1 November 1945.

The intuition of this approach is that stop words can produce noise affecting the coverage ratio hiding true entailment pairs to our recognition system. This can be seen in the following example:

- **T**: After playing, the dog sat on the mat.
- **H**: A dog sat over a mat.

Here the coverage ratio is $\frac{3}{6} = 0.5$ that could be marked by the system as a *false* textual entailment pair if the threshold was greater than 0.5, but we can see that this is a true textual entailment pair.

Now, lets remove the stop words in the text and the hypothesis:

- **T'**: playing, dog sat mat.
- **H'**: dog sat mat.

where the coverage ratio is $\frac{3}{3} = 1$, showing that this is a *true* textual entailment pair, like it was supposed to be.

Preprocessing: Stemming. The process of stemming consists in deleting the non essential part of the words such as suffixes and prefixes in order to obtain the essential part or *stem* of it. For example in the words **engineer**ing, **engineer**ed and **engineer** the essential part or stem of the words is **engineer**.

Stemming is used to improve retrieval effectiveness because it allows to match words that are not directly identical but their stem is the same. There are several stemming algorithms [15], each one performing the stemming task in different ways, but one of the most popular stemming algorithms for English is Porter's algorithm [18], the one used in this work.

Our preprocessing module that handles stemming takes as an input a natural language text fragment and returns it with the stem representation of each word. For example:

Input: Bountiful arrived after war's end, sailing into San Francisco Bay 21 August 1945. Bountiful was then assigned as hospital ship at Yokosuka, Japan, departing San Francisco 1 November 1945.
Output: Bounti arriv after war' end, sail into San Francisco Bai 21 August 1945. Bounti wa then assign as hospit ship at Yokosuka, Japan, depart San Francisco 1 Novemb 1945.

The intuition of this approach is that we can match words that are not directly the same, but that share the same stem so that they are related in some way and can uncover disguised relationships. Consider the following example:

– **T**: After eating and playing with the kids, the doggy started sleeping.
– **H**: The dog played.

Here the coverage ratio is $\frac{1}{3} = 0.33$, that is a low coverage ratio considering that this is a true textual entailment pair.

Now, lets stem both text and hypothesis:

– **T'**: After eat and plai with the kid, the doggi start sleep.
– **H'**: The dog plai.

that overcomes the previous coverage ratio with $\frac{2}{3} = 0.66$ that is closer to the correct answer.

Preprocessing: Negation Detection. Negation is present in all languages and in most cases statements are affirmative by default. Negation is used to change the polarity of the statements and typically denotes something unusual or exceptions. At first glance, negation seems easy to deal with because the problem can be thought as the task of simply inverting the polarity of the items covered by the scope of negation; however, this is not always the case.

Unlike affirmative statements, negation is marked by words (*not, no, never*) or affixes (*n't,un-*) and also connective adjuncts can be used to negate positive clauses, such as *neither* and *nor*. Another words that indicate negation are *nobody, none, nowhere*, etc.

There are two levels for handling the negation problem: lexical and syntactic.

The lexical approach uses the shallow representation of the sentences to detect negation, according to [3] *not* and *n't* correspond to 79.61% of negative bearing word occurrences based on the WSJ Penn Treebank.

The syntactic approach tries to discover the negative polarity of sentences by looking for patterns of negation based on a syntactic constituents parse tree, but this method requires using a syntactic parser.

In this work we use the detection of negation in a naïve way by searching for the occurrences of the word *not* and words that end with *n't*. Once found, we negate each word prepending **not_** until we reach either a comma or a period.

Our preprocessing module that handles negation takes as input a text fragment and returns the negative representation of it, for example:

Input: The British government did not initially purchase the weapon and civilian sales were modest. However the U.S. Civil War began in 1860 and the governments of both the United States and the Confederacy began purchasing arms in Britain.

Output: The British government did **not_initially not_purchase not_the not_weapon not_and not_civilian not_sales not_were not_modest**. However the U.S. Civil War began in 1860 and the governments of both the United States and the Confederacy began purchasing arms in Britain.

where we add the prefix *not_* to each word that is covered by the scope of the negation.

The intuition of preprocessing negation is that sometimes the system detects false entailment pairs because it does not consider the polarity of the expressions. An example of this is the following:

- **T**: The new car is fast but is not equipped with mp3 player.
- **H**: The new car is equipped with mp3 player.

As we can see in the previous example, the coverage ratio is $\frac{8}{8} = 1$, which means that independently of the selected threshold, the system would incorrectly mark these T and H as a *true* textual entailment pair.

Lets use now the negation detection technique just described, we would have then the following **T'** and **H'**.

- **T'**: The new car is fast but not_is not_equipped not_with not_mp3 not_player.
- **H'**: The new car is equipped with mp3 player.

After the preprocessing, the coverage ratio is $\frac{4}{8} = 0.5$ that helps the system to know that this may not be a true textual entailment pair.

2.2 Syntactic Module

Once we have discussed the lexical module, we proceed to explain the functionality of our syntactic module which operates at a deeper level of the input expressions **T** and **H**; that is, the syntactic level aims to model how the words of a sentence depend on each other. For this purpose, a previous parsing of T and H is required to obtain a *dependency syntactic parse tree* or a *constituent syntactic parse tree*.

The *dependency tree* shows dependencies between the words of an expression. Each edge is labeled with the dependency of the words that it connects, Figure 1 is an example of this kind of tree. The *constituent tree* indicates how the words of a sentence are grouped in lexical constituents that conform the whole expression in a hierarchy, as shown in Figure 2.

For this module we used the *Stanford syntactic parser* [13] which is able to produce both kind of syntactic parse trees. For dependency trees Stanford parser uses the following notation:

Subj(is, Gaspe)
Det(Gaspe, the)
Obj(is, peninsula)
Det(peninsula, a)

where each line represents an edge of the dependency tree. Figure 1 shows the graphic representation of this particular example.

For constituent trees, Stanford parser uses the following notation:

```
(ROOT
  (S
    (NP (DT the) (NNP Gaspe))
    (VP (VBZ is)
      (NP (DT a) (NN peninsula)))))
```

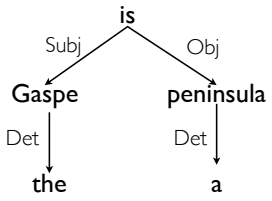


Fig. 1. Example of dependency syntactic parse tree

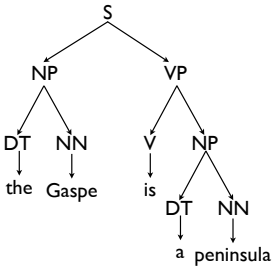


Fig. 2. Example of constituent syntactic parse tree

that corresponds to the tree shown in Figure 2. The general algorithm used in this module measures the coverage ratio of the edges in H's syntactic tree with regard to the syntactic parse tree of T. This means that, the more edges covered in the hypothesis, the more likely will be that it is a textual entailment pair.

In the same way that the lexical approach, we use a coverage threshold **TH** to decide if a given pair is or is not a textual entailment pair.

Both kind of parse trees (dependency and constituent) can be combined as will with complementary techniques like stop words, stemming, and negation, just described in the previous section.

2.3 Syntactic n-grams

Some of the most popular Natural Language Processing techniques are *n*-grams which are sequences of elements as they appear in the texts. The sequence of elements can be composed by words, characters, part of speech tags (POS), etc. The *n* term corresponds to number of elements to be considered by the sequence of elements. For example, the input expression:

– The small funny dog barks,

has the following 2-grams (bigrams): *the small, small funny, funny dog* and *dog barks*, and the following 3-grams (trigrams): *the small funny, small funny dog* and *funny dog barks*, and so on.

In this work we use syntactic n-grams (sn-grams), which are sequences of words that are obtained from the elements appearing in the syntactic trees (dependency or constituent trees) of a sentence. More specifically, sn-grams are constructed by the sequence of nodes that can be reached on any path of length n in the parse tree, namely, this kind of syntactic n-grams is known as continuous syntactic n-grams. In the remainder of this paper we will refer the continuous syntactic n-grams just as syntactic n-grams or sn-grams. There are several types of syntactic n-grams based on the types of elements they take into account:

- **Word sn-grams**: the elements of sn-grams are words.
- **POS sn-grams**: the elements of sn-grams are POS tags.
- **Syntactic relations sn-grams**: the elements of the sn-grams are names of syntactic relations between words.
- **Mixed sn-grams**: they are composed by mixed elements, like words, POS tags and/or syntactic relation types.

The main advantage of sn-grams is that they are based on syntactic relations of words, and thus, each word is bound to its "real" neighbors, ignoring the arbitrariness that is presented in the surface structure [22].

For example, the expression "the small funny dog barks" has the dependency syntactic parse tree illustrated in Figure 3, from which we can obtain the following syntactic bi-grams: *barks dog, dog the, dog funny* and *dog small*, and the following syntactic trigrams: *barks dog the, barks dog funny* and *barks dog small*.

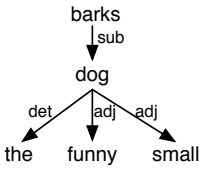


Fig. 3. Example of syntactic n-grams

Syntactic n-grams Extraction In this section we describe the procedure we followed to obtain syntactic n-grams, specifically *s2-grams*, *s3-grams* and *s4-grams* used for our Paraphrase Recognition system.

We based all our procedure on the syntactic dependency trees generated by the Stanford syntactic parser [13].

Our extraction process consists of two main steps:

1. Parse a expression with the Stanford parser.
2. Use the dependency relations obtained from the parse tree to form syntactic 2-grams, s3-grams and s4-grams.

For Step 1, we feed the Stanford parser with an input sentence, for example: "Economic news have little effect on financial markets." then we have as outcome the corresponding syntactic dependency tree, in this example is the following:

nn(news, Economic)
nsubj(have, news)
root(ROOT, have)
amod(effect, little)
dobj(have, effect)
amod(markets, financial)
prep_on(effect, markets)

this observes the notation for syntactic dependency trees defined by the Stanford parser. The above lines are the representation of the tree shown in Figure 4.

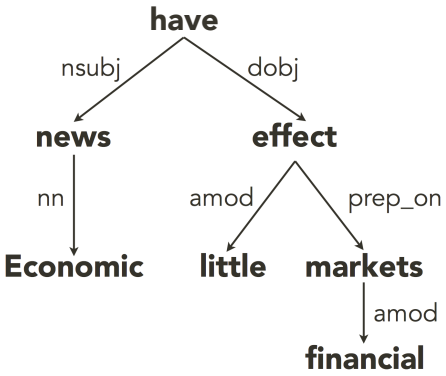


Fig. 4. Dependency tree for the sentence *Economic news have little effect on financial markets*

Syntactic 2-grams Extraction Using the parse tree corresponding to an expression, it is straightforward to obtain the word sn-grams by removing the label of the relation dependency of each edge of the tree, that is:

(news, Economic)
(have, news)

(ROOT, have)
(effect, little)
(have, effect)
(markets, financial)
(effect, markets)

are the s2-grams from the previous example.

Syntactic 3-grams Extraction Once that we have obtained the s2-grams, we can use them to conform the syntactic 3-grams by concatenating two s2-grams "a" and "b" if the second element of "a" is the first element of "b", for example *(have, news)* and *(news, Economic)* conform the s3-gram *(have, news, Economic)*. For the previous example we have the following s3-grams:

(have, news, Economic)
(ROOT, have, news)
(ROOT, have, effect)
(have, effect, markets)
(have, effect, little)
(effect, markets, financial)

Syntactic 4-grams Extraction Based on the syntactic 2-grams and 3-grams, we obtained the syntactic 4-grams. Given a syntactic 3-gram "a" and a syntactic 2-gram "b", we have the syntactic 4-gram "c" if the last element of "a" is the first element of "b", for example for the s3-gram *(have, effect, markets)* and the s2-gram *(markets, financial)* we can conform *(have, effect, markets, financial)* as syntactic 4-gram. For the previous example we have then:

(ROOT, have, news, Economic)
(ROOT, have, effect, little)
(ROOT, have, effect, markets)
(have, effect, markets, financial)

2.4 Semantic Module

A third approach that we experimented with is based on the *semantic level*. The semantics treatment is a very complex task that still has not been completely solved; however, our work has a shallow approach to it.

We are located in *linguistic semantics* that is a subfield of semantics; and linguistics, that studies the meaning of linguistic expressions that not depend on the context. More specifically, we use *Lexical Semantics* that is, the study of relations that exist between words where each one is seen like a node in a graph or a hierarchy, from which is possible to obtain *semantic relations* between them. For example, relations between

hypernyms, hyponyms, synonymy, antonymy, meronymy, and holonymy. In this work we only use the *hypernym* semantic relation and Lin's similarity measure [14] obtained from WordNet.

Preprocessing: Hypernym Representation. Sometimes the direct overlap between words of the given text and hypothesis is not possible; however, it could be the case that two words do not match directly but they match indirectly by means of a semantic relationship like hypernymy, hyponymy, etc. For example, the words *dog* and *canine* are not lexically matched, but using the hypernym relationship we can indirectly associate them, therefore the system would be able to detect "hidden" matches. We based our semantic module in the same algorithm that the lexical module uses; the difference is in the preprocessing step of T and H. This is based on the hypernym semantic relation that takes a text fragment as input and returns the hypernym representation for it. For example:

Input: Female mosquitoes become infected with the malaria parasite when they draw blood from humans with malaria.
Output: Animal dipterous insects become infected with the protozoal infection organism when they gully liquid body substance from homo with protozoal infection.

The intuition for this approach is that a more general representation of the concepts of T and H can be used for looking matches between words with the original T and H fragments, allowing the system to discover matches between words that were hidden before. For example, considering the following input pair:

- **T**: Every one knows that every canine hates felines.
- **H**: Every dog hates cats.

Here the coverage ratio is $\frac{2}{4} = 0.5$, which is marked by the system as a *false* textual entailment pair (the threshold was set to 0.5), but we can see that this is a true textual entailment pair. Now, lets apply the hypernym preprocessing module in the hypothesis:

- **T**: Every one knows that every canine hates felines.
- **H'**: Every canine hates felines.

where the coverage ratio is $\frac{4}{4} = 1$, showing that this is a *true* textual entailment pair, like it was supposed to be.

Semantic Approach with Lin's Similarity Measure. One special situation found in the lexical algorithm described earlier is presented when two words are not equivalent nor they are matched by the hypernym relationship, but they can be matched indirectly. For example, consider the words *spoon* and *fork* that are not directly equivalent but are closely related.

The relatedness treatment of words can help in the Textual Entailment recognition task allowing to recognize hidden matches between words that are not equivalent but are related.

The way on which the similarity or relatedness of words is measured is called *similarity measure*. There are several similarity measures but one of the most used ones is Lin's similarity measure [14] that is defined as:

$$Sim_{lin} = \frac{2 \times IC(LCS)}{IC(concept_1) + IC(concept_2)}$$

which is based on the information content (IC) of two concepts under the WordNet hierarchy.

The modified recognition algorithm based on the similarity measure is as follows:

Given a pair of expressions T, H a threshold TH and a similarity threshold STH:

$T \leftarrow$ preprocessing(T)
$H \leftarrow$ preprocessing(H)
$L_H \leftarrow$ length(H)
$common \leftarrow 0$
for all word w in H **do**
  for all word m in T **do**
    if $equals(w, m)$ **then**
      $common \leftarrow common + 1$
    else
      if $LinSimilarity(w, m) \geq STH$ **then**
        $common \leftarrow common + 1$
      end if
    end if
  end for
end for
$coverage \leftarrow common/L_H$
if $coverage \geq TH$ **then**
  return "YES"
else
  return "NO"
end if

As can be seen, the similarity measure algorithm is almost the same as the general lexical algorithm, with the difference that if a direct match cannot be found for a given word "m" of the hypothesis, its similarity with each word of the text is calculated, and if it is larger than a specified similarity threshold (*THS*), then the pair is considered as a match.

Additionally we use the combination of the preprocessing modules defined in the lexical section in order to improve the performance of the system.

3 Experiments for Paraphrase Recognition

The development of this section is focused on the Paraphrase Recognition task. The evaluation and tests will be based on the Microsoft Research Paraphrase Corpus (MSRP)

that consists in 5,801 pairs of paraphrase candidates which are divided in two sets: *test* and *training*.

The training set is composed of 4,076 pairs, whereas the test set has 1,725 pairs. Each pair was submitted to human judges who assigned a one label of 1 when it is a true paraphrase pair and 0 otherwise. An example of a pair is the following:

```
1 702876 702977 Amrozi accused his brother, whom he called ``the
witness'', of deliberately distorting his evidence. Referring to
him as only ``the witness'', Amrozi accused his brother of
deliberately distorting his evidence.
```

The Paraphrase Recognition task consists in, given two expressions S_1 and S_2, decide if they are a paraphrase of each other. The following sections describe the process and evaluation of our paraphrase recognition system based on syntactic *n*-grams.

3.1 Recognition Process

Our experiments are based on the idea that syntactic n-grams can provide more information than classic *n*-grams because they work at a deeper level considering the relations and dependencies between words of the input expression.

Syntactic *n*-grams can also be useful for detecting the "real" neighbors of each word ignoring the arbitrariness that is presented on the surface structure such as adjectives before nouns. For example in the previous sentence "the small funny dog barks," we can do the following comparison:

Table 1. n-grams vs. syntactic n-grams.

2-grams	**Syntactic 2-grams**
the small	barks dog
small funny	dog the
funny dog	dog funny
dog barks	dog small

As can be seen in Table 1, traditional 2-grams produce pairs that convey less information like "small funny" or "the small", whereas the syntactic 2-grams are all meaningful pairs.

We experimented with five different approaches for the recognition process:

1. Overlapping Syntactic n-grams.
2. Overlapping Syntactic n-grams and stemming.
3. Overlapping Syntactic n-grams and synonym detection.
4. Overlapping Syntactic n-grams and Lin's similarity measure.
5. Overlapping Syntactic n-grams and linear interpolation.

As in Textual Entailment, the general process for the Paraphrase Recognition system independently of the implemented approach, consists in measuring the differences between the sn-grams of the input expressions $\mathbf{S}_1$ and $\mathbf{S}_2$. A difference threshold **T** is used to decide if the input expressions are or not paraphrases of each other. That is, if the difference between them is less than the threshold T, they are paraphrases; otherwise they are not paraphrases. The algorithm is presenting below:

Given two sentences S_1 and S_2 and a threshold of difference **T**:

$L_1 \leftarrow$ sn-grams(S_1)
$L_2 \leftarrow$ sn-grams(S_2)
if $size(L_1) \geq size(L_2)$ **then**
  $T' \leftarrow size(L_1) \times T$
else
  $T' \leftarrow size(L_2) \times T$
end if
$D \leftarrow$ sn-gramsDifferences(L_1, L_2)
if $D < T'$ **then**
  return "YES"
else
  return "NO"
end if

In the next subsections we describe each of the approaches used for our recognition system, along with the result obtained with each one of them.

We use the *Microsoft Research Paraphrase Corpus* to evaluate the accuracy, precision, recall and F-measure scores of our system. With these values is possible to compare the performance achieved by our Paraphrase Recognition system with other works of the state of the art.

The Microsoft Research Paraphrase Corpus provides two datasets: *train* and *test*. With the train set we adjust the optimal value for our difference threshold to be the one that produces the best scores. Once that we decided the optimal threshold, we use it to evaluate the system with the test set. According to our experiments, the best threshold for our system is **0.85**.

3.2 Overlapping Syntactic N-grams

The simplest approach implemented in our system, is just to count the overlapping syntactic n-grams in the two input expressions S_1 and S_2. We experimented with s2-grams, s3-grams and s4-grams independently.

We repeated each experiment adjusting the maximum difference threshold in the range of 0.2 to 0.9. This approach is based on the general process algorithm. For instance, lets suppose that we are given the following input expressions under a threshold **T** of 0.3:

S_1: A mathematician solved the problem.

S_2: The problem was solved by a mathematician.

applying our s2-grams extraction module we obtain the s2-grams shown on Table 2 for S_1 and S_2 respectively.

Table 2. Syntactic 2-grams corresponding to S_1 and S_2.

s2-grams for S_1	**s2-grams for S_2**
(mathematician, a)	(problem, the)
(solved, mathematician)	(solved, problem)
(ROOT, solved)	(solved, was)
(problem, the)	(ROOT, solved)
(solved, problem)	(mathematician, a)
	(solved, mathematician)

Results. For this example, we can see that T' can be computed from Table 2: **T'** = size(L_2) $\times$ T = 6 $\times$ 0.3 = **2**. The remaining process consist on determining if S_1 and S_2 do not differ by more than 2 words. It can be seen that S_1 and S_2 differ only on the s2-gram *(solved, was)*, therefore this would be considered as a *true* paraphrase pair by the system. The same procedure is applied individually for s3-grams and s4-grams.

Table 3 shows results obtained when we use the basic technique of common syntactic n-grams for syntactic 2-grams, 3-grams and 4-grams.

As we can see in the table, our system achieves the highest F-measure of 80.3% when*s2-grams* are used. However it can also be seen that the higher order of the syntactic n-grams, the higher precision value is obtained. On the other hand, the higher order of syntactic n-grams, the less recall value is obtained.

Table 3. Results of syntactic n-grams approach

	Accuracy	**Precision**	**Recall**	**F-measure**
S2-grams	**68.3%**	**68.3%**	**97.4%**	**80.3%**
S3-grams	64.2%	72.8%	73.6%	73.2%
S4-grams	56.9%	74.4%	53.7%	62.4%

3.3 Overlapping Syntactic N-grams and Stemming

As explained in Section 2.1 for Textual Entailment, Stemming is a process that removes the non essential part of the words such as suffixes and prefixes in order to obtain the *essential* part or stem of a word. For example in the words **fish**ing, **fish**ed, **fish**er the stem is **fish**.

We use the Porter stemming algorithm [18], since it is the most popular stemming algorithm for English, but there are many others.

Our sn-grams post processing module that works with stemming, takes as input a syntactic n-gram and returns the stemmed representation of it, for example:

- *(redness, car)* $\rightarrow$ *(red, car)*
- *(fully, car, engineered)* $\rightarrow$ *(fulli, car, engin)*

The Paraphrase Recognition process based on stemming performs the following steps:

1. Obtain sn-grams for each input expression.
2. Apply stemming for each sn-gram obtain in the previous step.
3. Use the general algorithm.

With this approach we intent to recognize syntactic n-grams that can not be related in a direct way but that may be indirectly related by means of their stems.

Results of Overlapping Syntactic N-grams and Stemming. Now we show in Table 4 the achieved results for the system when post processing of the syntactic n-grams is used, by transforming each one on its stemmed representation.

As the table shows, the stemming of syntactic n-grams slightly increases the general performance of the system, yielding an increase in accuracy, precision, recall and the F-measure.

The F-measure obtained with this approach is 80.6% by using *s2-grams*. Again the precision increases and the recall decreases when a higher degree of syntactic n-gram is used.

Table 4. Results of syntactic n-grams with stemming approach

	Accuracy	**Precision**	**Recall**	**F-measure**
S2-grams	**68.6%**	**68.4%**	**97.9%**	**80.6%**
S3-grams	64.5%	72.5%	74.9%	73.7%
S4-grams	57.8%	74.4%	55.7%	63.7%

3.4 Overlapping Syntactic N-grams and Synonym Detection

One way of finding "hidden" related pairs of words in sentences or related pairs of syntactic n-grams in the paraphrase recognition process is by using synonyms. For instance, suppose the syntactic 2-grams: **(car, red)** and **(automobile, carmine)**: our basic paraphrase recognition system would not match this pair, but it can be seen that they should be considered as a match.

In order to cope with this kind of problems, we introduce a synonym detection module based on WordNet synsets that are considered as groups of synonym words.

Two words are synonyms if they can be interchanged under the same scope without modifying the truth value of an expression, in other words, for our purpose, synonyms are words that are considered equivalent. For example, the words *plant* and *flower*, *home* and *house*, *kid* and *child* can be considered as equivalent.

Applying this intuition to our Paraphrase Recognition system, a post processing module was developed. This module takes a syntactic n-gram as input and returns four equivalent syntactic sn-grams by obtaining the most common synonym of each word from WordNet.

Taking the first example of this section, if we apply our module to *(car, red)*, we obtain the following:

- (car, red)
- (auto, red)
- (car, redness)
- (auto, redness)

This answer is conformed by the combination of the synonyms of the words *car* and *red*, the combination is required because we need to compare each possible variation of the pair, augmenting the recognition capability of the system.

The result of applying the same procedure to *(automobile, carmine)* is:

- (automobile, carmine)
- (auto, carmine)
- (automobile, red)
- (auto, red)

From the previous two answers the system now is able to detect the match between *(car, red)* and *(automobile, carmine)* because both share the same synonym form *(auto, red)*.

A similar process is used for syntactic 3-grams and 4-grams treatment, the only difference is that we obtain 9 and 16 synonym combinations respectively.

Results of Overlapping Syntactic N-grams and Synonyms. Table 5 shows the system performance when we apply synonym postprocessing of sn-grams. This time the synonym approach helps to improve the basic approach (direct overlapping), but it does not contribute as much as stemming.

The highest F-measure is 80.4% with *s2-grams* again.

Table 5. Results for the synonym approach

	Accuracy	Precision	Recall	F-measure
S2-grams	**68.4%**	**68.3%**	**97.7%**	**80.4%**
S3-grams	66.4%	66.9%	97.6%	79.4%
S4-grams	65.4%	67%	94.5%	78.4%

3.5 Overlapping Syntactic N-grams and Lin's Similarity Measure

Another related problem found on the process of recognizing a match between syntactic n-grams is presented when two elements are not directly equivalent and also not matched by using the synonym technique described in the previous section. An example is the following pair of syntactic 2-grams: **(song, romance)** and **(music, love)**. In this case there is not a direct match between words and, if we consider the synonyms shown on the Table 6, we can see that no match can be found. For this reason we use

Table 6. Example: Synonym approach.

Synonyms (song, romance)	**Synonyms (music, love)**
(song, romance)	(music, love)
(vocal, romance)	(euphony, love)
(song, romanticism)	(music, passion)
(vocal, romanticism)	(euphony, passion)

an additional approach based on word similarity measures. More specifically, we use the same similarity measure we used for Textual Entailment shown in Section 2.4. For reader's convenience, we reproduce here the formula for its calculation: [14]:

$$Sim_{lin} = \frac{2 \times IC(LCS)}{IC(concept_1) + IC(concept_2)}$$

Using this measure we try to detect similar syntactic n-grams by considering two corresponding words as a match if their similarity measure is greater than, or equal to, a threshold value.

The Lin similarity measure between *song* and *music* is 0.86 and between *romance* and *love* is 0.53; thus, if the threshold of similarity were 0.5, the pair of syntactic 2-grams (*song, music*) and (*romance, love*) would be considered as a match.

Overlapping Syntactic N-grams and Lin Similarity. Now we present the results obtained by the system when we use a comparison technique between sn-grams with Lin's similarity measure, instead of directly comparing them.

Table 7 shows that the performance of the system decreases in accuracy, precision, recall and F-measures with regard to the previous shown approaches. The highest F-measure obtained is 80.1%, again with the syntactic *s2-grams*.

3.6 Overlapping Syntactic N-grams and Linear Interpolation

An additional approach that we tried in order to improve the system performance is based on the linear interpolation technique used in traditional n-grams that consists in creating a linear interpolation of the syntactic 4-grams, 3-grams and 2-grams models, each one weighted by a λ value, where the sum of all lambdas must be equal to 1.

Table 7. Results of syntactic n-grams with Lin's similarity

	Accuracy	**Precision**	**Recall**	**F-measure**
S2-grams	**67.4%**	**67.4%**	**98.7%**	**80.1%**
S3-grams	61.1%	72.1%	77.3%	74.6%
S4-grams	57.3%	73.3%	56.4%	63.8%

We think that overlapping s4-grams mean a stronger relationship between two sentences compared with s3-grams. We considered the same idea between s3-grams and s2-grams, therefore, we assigned greatest weight to s4-grams, then a smaller weight for s3-grams and the lowest weight was assigned to s2-grams. The intuition here is that the higher the degree of overlapping n-grams, the more similar two texts are.

We represent this idea as follows:

$$F_S = \lambda_4 \times snGrams(S_1, S_2, 4) + \lambda_3 \times snGrams(S_1, S_2, 3) + \lambda_2 \times snGrams(S_1, S_2, 2)$$

Where snGrams(S_1, S_2, n) represents the common syntactic n-grams of degree *n* between the expressions S_1 and S_2 and $\lambda_4 + \lambda_3 + \lambda_2 = 1$.

Consider the following example:

- $\mathbf{S}_1$: A mathematician solved the problem.
- $\mathbf{S}_2$: The problem was solved by a mathematician.

As we can see from the Table 8 the two input expressions do not have overlapping syntactic 4-grams, therefore our system based on s4-grams would mark the expressions as a *false* paraphrase pair; however, it can be seen that by applying linear interpolation we are able to obtain an score grater than zero, allowing the system to consider a different answer. The computation using this approach is:

$$F_S = 0.5 \times 0 + 0.3 \times 2 + 0.2 \times 4 = 1.4,$$

where $\lambda_4 = 0.5, \lambda_3 = 0.3, \lambda_2 = 0.2$.

The obtained value is then normalized to get a final score between 0 and 1 that can be used together with a similarity threshold to decide if the expressions are or are not a true paraphrase pair.

Results of Overlapping Syntactic N-grams and Linear Interpolation. Now we present in Table 9 the results obtained for the system when we use the linear interpolation approach. As can be seen, the system performance decreases considerably when using Linear Interpolation; in part, due to the fact that the lambda values require an exhaustive procedure to obtain an optimal value for them. We experimented with several lambda values with low success.

Table 8. Linear interpolation of syntactic n-grams

n	**Input 1**	**Input 2**	**snGrams(S_1, S_2, n)**
2	(mathematician, a) (problem, the) (solved, problem) (solved, mathematician)	(mathematician, a) (problem, the) (solved, problem) (solved, mathematician) (solved, was)	4
3	(solved, mathematician, a) (solved, problem, the)	(solved, mathematician, a) (solved, problem, the)	2
4	No	No	0

For this particular approach we used a threshold of 0.2. Note that in this case the algorithm is based on similarity, contrasting with the other approaches that are based on distance.

Table 9. Results with linear interpolation

$\lambda_4, \lambda_3, \lambda_2$	**Accuracy**	**Precision**	**Recall**	**F-measure**
0.5, 0.3, 0.2	60.1%	75.1%	59.8%	66.6%
0.4, 0.3, 0.3	**62.9%**	**74.6%**	**67%**	**70.6%**
0.7, 0.2, 0.1	57.8%	76.6%	52.6%	62.4%

The best F-measure shown in Table 9 is 70.6%, with $\lambda_4 = 0.4, \lambda_3 = 0.3$ and $\lambda_2 = 0.3$.

4 Experiments for Textual Entailment

In this section we describe and compare the experiments for the different approaches described in Section 2 and their combinations used by our system. The evaluation of this system is based on the PASCAL RTE-3 dataset. We used the development set for calibrating the best threshold value and then we used the test set for the final evaluation. The system evaluation is based on the accuracy, precision, recall and the F-measure score; however we focus mainly on the *accuracy* measure, because that is the one used in the RTE challenge to compare results.

4.1 The Lexical Approach Results

The Table 10 shows the scores obtained by our system with the lexical approach, it shows the basic approach (simple overlapping) and all the combinations of complementary techniques used. Notice how the *stemming* and *removing stop words* techniques contribute in both ways, individually and together to improve the performance of the base system that does not use any additional techniques.

An important observation is that apparently the negation treatment not only does not help to improve the results but also affects it.

We conclude that the best combination under the auxiliary techniques that we used for the lexical module is using *stemming* and *removing stop words* together, reaching and accuracy of 66% and precision of 61.5%, with a threshold of 0.5.

Table 10. Lexical results

Stemming	Stop words	Negation	Accuracy	Precision	Recall	F-measure	Threshold
			64.3%	62.8%	74.3%	68.1%	0.65
X			64.7%	61.5%	82.9%	70.6%	0.65
	X		65.2%	62.2%	81.9%	70.7%	0.55
		X	61.1%	61.4%	64.6%	63.0%	0.50
X	**X**		**66.0%**	**61.5%**	**89.7%**	**73.0%**	**0.50**
	X	X	65.0%	62.0%	81.4%	70.4%	0.55
X		X	64.2%	62.9%	73.4%	67.7%	0.70
X	X	X	65.6%	61.2%	89.5%	72.7%	0.50

4.2 The Syntactic Approach Results

Now we discuss the syntactic approach results. Our intuition was that the deeper the level of analysis, the more accurate is the system; however, by using a basic syntactic approach, we got lower results; maybe because a more sophisticated syntactic technique is required.

Table 11 shows the achieved results. For each experiment we determined the best threshold. We first tried a simple edge overlapping technique; then we tried splitting both T and H by periods with the idea of creating a more accurate parsing. We also present the results of using constituent trees and the auxiliary techniques applied after the parsing step.

The best score achieved is an accuracy of 58.5% and precision of 56.8% under a threshold of 0.1.

Table 11. Syntactic results

Overlapping Edges	Accuracy	Precision	Recall	F-measure	Threshold
Simple	57.0%	56.4%	70.0%	62.5%	0.05
Splitting by periods	57.8%	56.8%	73.6%	64.1%	0.05
Constituents	54.2%	52.9%	95.1%	68.0%	0.30
Stop words	57.3%	59.0%	55.1%	56.9%	0.15
Negation	56.8%	57.0%	64.1%	60.3%	0.05
Stemming	**58.5%**	**56.8%**	**79.2%**	**66.1%**	**0.10**

4.3 The Semantic Approach Results

Table 12 shows the results achieved by our semantic approach. Conversely to the lexical approach, negation treatment gives some benefit, but not enough to reach a higher score. Note also that in this case, word similarity does not contribute to overcome the results. The best combination was *hypernym, stemming* and *stop words* altogether, with an accuracy of 64.0% and a precision of 61.0% with a threshold of 0.65.

Table 12. Semantic results

Technique	Accuracy	Precision	Recall	F-measure	Threshold
Hypernym	62.6%	63.7%	62.6%	63.2%	0.70
Hypernym and stemming	62.2%	59.6%	81.2%	68.8%	0.60
Hypernym and stop words	62.6%	60.1%	80.4%	68.8%	0.55
Hypernym, stop words and word sim.	60.3%	58.2%	80.0%	67.4%	0.80
Word similarity	55.3%	53.6%	94.1%	68.3%	0.10
Word similarity and stemming	54.3%	53.0%	96.0%	68.3%	0.10
Word similarity and stop words	60.0%	60.6%	62.6%	61.6%	0.25
Word similarity, stop words and stemming	60.5%	60.5%	65.6%	62.9%	0.25
Hypernym, stop words and negation	62.5%	60.1%	79.7%	68.5%	0.55
Hypernym and negation	63.5%	61.0%	79.5%	69.0%	0.65
Hypernym, stemming and stop words	**64.0%**	**61.0%**	**81.9%**	**70.0%**	**0.65**

5 Conclusions and Future Work

5.1 Paraphrase Recognition

Summarizing the result tables shown in Section 3, it can be seen that the best scores the system yields is an F-measure of **80.6%**, this is obtained by using the auxiliary post processing technique of stemming, applied to each syntactic n-gram obtained in the syntactic parse step.

The synonym technique is also a good approach since the performance of the system improved with regard to the basic algorithm, which consists on simple syntactic n-grams overlapping; however, it does not contribute as much as the stemming technique.

On the other hand the Lin's similarity measure and linear interpolation approaches that we applied do not seem to contribute for this specific task.

After experimenting with syntactic n-grams and some complementary techniques, we conclude that syntactic n-grams can be used successfully achieving good results in the Paraphrase Recognition task; however there is still a lot of room for improvement.

Table 14 shows the unsupervised reported scores using the Microsoft Research Paraphrase Corpus and we also show our results in order to compare them with these works.

As future work, a deeper analysis can be done, dealing on how to apply similarity measures to compare syntactic n-grams and also how to choose the optimal *lambda*

values for the linear interpolation approach. Another approach that remains to be tested is the use of non-continuous syntactic n-grams [21] since the syntactic n-grams used in this work were continuous syntactic n-grams.

Table 13. Unsupervised reported works

Author	Accuracy	F-measure
Fernando and Stevenson, 2008	74.1%	82.4%
Islam and Inkpen, 2007	72.6%	81.3%
Mihalcea et al., 2006	70.3%	81.3%
our system, 2013	**68.6%**	**80.6%**
Rus et al., 2008	70.6%	80.5%
Mihalcea et al., 2006	65.4%	75.3%

5.2 Conclusions for Textual Entailment

Comparing Tables 10, 11 and 12, we can see that the best result achieved by our system was obtained under the simple lexical approach with an accuracy of 66% and a precision of 61.5%. Although this is not the highest score compared with the state of the art for RTE-3, it shows that by using a few additional resources it is possible to obtain fair results. This is good news, because in some languages like *Tagalog*, complex resources could represent a restriction for Textual Entailment recognition systems.

In Table 14 we can see the top 8 reported results on RTE-3 compared with our system, showing that most of them use several auxiliary resources like theorem provers, knowledge bases, logical inference and so on.

Table 14. Resources used

Author	Accuracy	Precision	Lexical	Syntactic	Semantic	Logical Inference	Background knowledge	Machine Learning
Hickl	80%	88.15%	X		X	X	X	X
Tatu	72.25%	69.42%	X		X	X	X	
Iftene	69.1%	-	X	X			X	
Adams	67%	-	X				X	X
Zanzotto	66.7%	66.7%	X	X			X	
Wang	66.5%	-		X				X
Us	**66%**	**61.5%**	**X**					
Blake	65.8%	60.96%	X	X				X
Ferrandez	65.6%	-	X	X				

As future work, negation treatment can be improved, as well as the syntactic technique used for this work, taking advantage of more information Linguistics can provide.

References

1. Androutsopoulos, I. & Malakasiotis, P. (2010). "A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research*, 38 135-187.
2. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. & Szpektor, I. (2006). "The Second PASCAL Recognising Textual Entailment Challenge", *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*, Venice, Italy.
3. Blanco, E. & Moldovan, D. (2011). "Some Issues on Detecting Negation from Text", *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, Florida, Association for the Advancedment of Artificial Intelligence, 228-233.
4. Bos, J. & Markert, K. (2005). "Recognising Textual Entailment with Logical Inference", *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, October 2005. Stroudsburg, PA, USA: Association for Computational Linguistics, 628-635.
5. Dagan, I., Glickman, O. & Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B. & D'Alchébuc, F. eds. (2006). *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, pp. 177-190.
6. Das, D. & Smith, N. (2009). "Paraphrase Identification As Probabilistic Quasi-synchronous Recognition", Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, August. Stroudsburg, PA, USA: Association for Computational Linguistics, 468-476.
7. Dolan, B., Quirk, C. & Brockett, C. (2004). "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources", *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, Stroudsburg, PA, USA: Association for Computational Linguistics.
8. Dolan, W. & Brockett, C. (2005). "Automatically Constructing a Corpus of Sentential Paraphrases", *Proceedings of the 3rd International Workshop on Paraphrasing*, Jeju island, Korea, 9-16.
9. Giampiccolo, D., Magnini, B., Dagan, I. & Dolan, B. (2007). "The Third PASCAL Recognizing Textual Entailment Challenge", *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, Stroudsburg, PA, USA: Association for Computational Linguistics, 1-9.
10. Hickl, A. & Bensley, J. (2007). "A Discourse Commitment-based Framework for Recognizing Textual Entailment", *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, Stroudsburg, PA, USA: Association for Computational Linguistics, 171-176.
11. Jurafsky, D. & Martin, J. (2008). *Speech and language processing*, (2nd edition). Upper Saddle River, N.J.: Prentice Hall.
12. Kauchak, D. & Barzilay, R. (2006). "Paraphrasing for Automatic Evaluation", *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, June 2006. Association for Computational Linguistics, 455-462.
13. Klein, D. & Manning, C. (2003). "Accurate Unlexicalized Parsing", *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, Stroudsburg, PA, USA: Association for Computational Linguistics, 423-430.

14. Lin, D. (1998b). "An Information-Theoretic Definition of Similarity", *Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 296-304.
15. Manning, C., Raghavan, P. & Schtze, H. (2008). *Introduction to information retrieval*. New York: Cambridge University Press.
16. Mihalcea, R., Corley, C. & Strapparava, C. (2006). "Corpus-based and Knowledge-based Measures of Text Semantic Similarity", *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, Massachusetts, American Association for Artificial Intelligence, 775-780.
17. Miller, G. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, Iss. Nov. 1995 39-41.
18. Porter, M. (1997). An algorithm for suffix stripping. *Readings of Information Retrieval*, 313-316.
19. Qiu, L., Kan, M. & Chua, T. (2006). "Paraphrase Recognition via Dissimilarity Significance Classification", *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, Stroudsburg, PA, USA: Association for Computational Linguistics, 18-26.
20. Ros Gaona, M., Gelbukh, A. & Bandyopadhyay, S. (2010). "Recognizing Textual Entailment Using a Machine Learning Approach", *Proceedings of the 9th Mexican International Conference on Artificial Intelligence Conference on Advances in Soft Computing: Part II*, Pachuca, Mexico, Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 177-185.
21. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A. & Chanona-Hernández, L. (2013). "Syntactic Dependency-based N-grams: More Evidence of Usefulness in Classification", *Conference on Intelligent Text Processing and Computational Linguistics*, Springer-Verlag Berlin Heidelberg, 13-24.
22. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A. & Chanona-Hernández, L. (2012). Syntactic Dependency-based N-grams as Classification Features. In: Batyrshin, I. & González, M. eds. (2013). *Advances in Computational Intelligence*. Springer-Verlag Berlin Heidelberg, pp. 1-11.
23. Tatu, M. & Moldovan, D. (2006). "A Logic-based Semantic Approach to Recognizing Textual Entailment", *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, Sydney, Australia, Stroudsburg, PA, USA: Association for Computational Linguistics, 819-826.
24. Tatu, M. & Moldovan, D. (2005). "A Semantic Approach to Recognizing Textual Entailment", *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, British Columbia, Canada, Stroudsburg, PA, USA: Association for Computational Linguistics, 371-378.
25. Zanzotto, F., Pennacchiotti, M., & Moschitti, A. (2009). "A machine learning approach to textual entailment recognition." *Natural Language Engineering*, 15 (04), 551-582.

Parallel-Wiki: A Collection of Parallel Sentences Extracted from Wikipedia

Dan Ştefănescu[1,2] and Radu Ion[1]

[1] Research Institute for Artificial Intelligence, Romanian Academy
{danstef,radu}@racai.ro
[2] Department of Computer Science, The University of Memphis
dstfnscu@memphis.edu

Abstract. Parallel corpora are essential resources for certain Natural Language Processing tasks such as Statistical Machine Translation. However, the existing publically available parallel corpora are specific to limited genres or domains, mostly juridical (e.g. JRC-Acquis) and medical (e.g. EMEA), and there is a lack of such resources for the general domain. This paper addresses this issue and presents a collection of parallel sentences extracted from the entire Wikipedia collection of documents for the following pairs of languages: English-German, English-Romanian and English-Spanish. Our work began with the processing of the publically available Wikipedia static dumps for the three languages involved. The existing text was stripped of the specific mark-up, cleaned of non-textual entries like images or tables and sentence-split. Then, corresponding documents for the above mentioned pairs of languages were identified using the cross-lingual Wikipedia links embedded within the documents themselves. Considering them comparable documents, we further employed a publically available tool named LEXACC, developed during the ACCURAT project, to extract parallel sentences from the preprocessed data. LEXACC assigns a score to each extracted pair, which is a measure of the degree of parallelism between the two sentences in the pair. These scores allow researchers to select only those sentences having a certain degree of parallelism suited for their intended purposes. This resource is publically available at:
http://ws.racai.ro:9191/repository/search/?q=Parallel+Wiki

Keywords: Parallel Data, Comparable Corpora, Statistical Machine Translation, Parallel Sentence Extraction for Comparable Corpora

1 Introduction

During recent years, Statistical Machine Translation (SMT) has received a lot of attention from the scientific community, attracting more and more researchers. Some of this interest is due to companies like Google or Microsoft, whose public SMT engines attract a great deal of curiosity and shape the belief that building an SMT system for informative translations that is widely accepted by Internet users is very near (increasing the level of awareness of the field). However, much of the research in this direc-

tion makes use of the same SMT model (Shannon's noisy channel) with its very popular implementation, the Moses SMT Toolkit (Koehn et al., 2007). So far, Moses has been proven to be the best publically available engine on which SMT systems are built. The differences in quality between such systems come to depend on the resources used by the Moses decoder or on the post-processing steps which aim to correct some of its mistakes. Still, in terms of procuring some of the needed resources like translation models, the scientific community has very few publically available options even for resource-rich languages. In order to build translation models, one needs parallel text aligned at the sentence level and such resources cannot be easily acquired in large quantities. Most of the available ones are juridical, medical or technical collections of documents, which are often the result of efforts beyond the NLP field. For example, JRC-Acquis[1] (Steinberger et. al, 2006) is a well-known collection of juridical parallel texts in 22 languages covering the EU legislation. It is the most used parallel corpus for Statistical Machine Translation experiments. OPUS[2] (Tiedemann, 2012) is a collection of parallel corpora that includes many known freely available such resources. Some of them are: (i) EUROPARL (European Parliament Proceedings) (Koehn, 2005), (ii) EUconst (the European constitution), which are both juridical texts, (iii) EMEA (European Medicines Agency documents) which belongs to the medical domain, several technical parallel texts like (iv) ECB (European Central Bank corpus), (v) KDE4 localization files, (vi) KDE manual corpus, (vii) PHP manual corpus, etc., some subtitles corpora like (viii) OpenSubs or (ix) TEP (Tehran English-Persian subtitle corpus) and news corpora like SETIMES (parallel news corpus of the Balkan languages).

From the above enumeration of existing parallel texts, one can infer that the general domain is poorly covered and more than this, there are languages for which parallel texts are scarce, no matter the domain. This is why the research community started to explore the possibility of acquiring parallel data from comparable texts. Such texts contain documents referring to the same subject or topic, but are not reciprocal translations. The problem of extracting data from comparable corpora began to be studied in the late 90s, as soon as people realized that the Web can be seen as a vast source of comparable documents. Among the important contributions to this area of research we have to mention the works of Wu (1994), Zhao and Vogel (2002), Resnik and Smith (2003), Fung and Cheung (2004), Munteanu and Marcu (2005), Quirk et al. (2007) and Tillmann (2009). Recent research includes that of Rauf and Schwenk (2011) and Ştefănescu et al. (2012). The most recent European projects on this topic are ACCURAT[3] (Analysis and evaluation of Comparable Corpora for Under Resourced Areas of machine Translation) and TTC[4] (Terminology extraction, Translation tools and Comparable corpora).

Our own experiments on mining parallel data from comparable corpora were conducted within the ACCURAT project and led to the development of a tool named

[1] http://ipsc.jrc.ec.europa.eu/index.php?id=198
[2] http://opus.lingfil.uu.se/
[3] http://www.accurat-project.eu/
[4] http://www.ttc-project.eu/

LEXACC[5]. With LEXACC, which is thoroughly described in Ştefănescu et al. (2012), one can extract parallel sentences from comparable corpora even if the level of comparability is low.

This paper describes the process of using LEXACC for harvesting parallel sentences from Wikipidia[6], for three language pairs: English-German, English-Romanian and English-Spanish. The next section presents related work, the following one gives information about the Wikipedia data we considered and the pre-processing steps we undertook in order to clean it. Section 3 details the procedure we followed for extracting the data, while section 4 contains statistics about the newly created resources. The paper ends with conclusions and ideas for further research.

2 Related Work

Considering that the largest existing publically available database of comparable documents is Wikipedia, a natural step would be to use it for harvesting parallel data. Adafre and Rijke (2006) are among the first to follow this idea, working on English-Dutch pair of languages. They suggested two approaches. The first one employs an MT system to generate a rough translation of a page and then uses word overlap between sentences as a similarity measure. In the second approach, the similarity measure is likewise computed, but this time the sentences are represented by entries in a shared lexicon built on concepts and entities that have entries in Wikipedia. Adafre and Rijke conducted small-scale experiments on a random sample of 30 Wikipedia page pairs. To find the parallel sentences, they considered the entire Cartesian product of source-target sentence pairs, an approach which is not feasible when dealing with datasets many orders of magnitude larger.

Yasuda and Sumita (2008) proposed a framework of a Machine Translation (MT) bootstrapping method on multilingual Wikipedia articles. According to the authors, this method can "simultaneously generate a statistical machine translation and a sentence-aligned corpus." They conducted their experiments on the Japanese-English language pair, working with a 2007 Wikipedia version. As stated by the authors, at that time, the on-line encyclopedia contained 1,500 sentences for Japanese and 50,000,000 for English and they also considered the entire Cartesian product of these sets in the process of finding similar sentence pairs.

Smith at al. (2010) exploited the observation that parallel sentences pairs are frequently found in close proximity and attempted to model the document level alignment accordingly. To do that, they used four categories of features: (i) features derived from word alignments, (ii) distortion features, (iii) features derived from Wikipedia markup and (iv) word-level induced lexicon features. Smith at al. worked on three language pairs (i.e. Spanish-English, Bulgarian-English and German-English), conducting small-scale experiments on 20 Wikipedia article pairs for each language pair. While their datasets are no longer available at the URL provided within the paper, their work showed that Wikipedia is a useful resource for mining parallel data.

[5] http://nlptools.racai.ro/nlptools/index.php?page=lexacc
[6] http://www.wikipedia.org/

Mohammadi and Ghasem-Aghaee (2010) improved the method of Adafre and Rijke (2006) by making use of Gale and Church (1993) observation that "longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences." Consequently, they reduced the search space represented by the Cartesian product between the sets containing the source and target sentences within an article pair. They experimented with different similarity measures between candidate pairs: Dice, Cosine and Jaccard coefficients, the latter obtaining the best results. Mohammadi and Ghasem-Aghaee evaluated their method on 30 Wikipedia article pairs and constructed a Persian-English parallel corpus by mining 1,600 article pairs.

Another work we have to mention is that of Birch et al. (2011) who released the Indic multi-parallel corpus in December, 2011. This corpus contains about 2,000 Wikipedia sentences translated into 6 Indic languages. As mentioned by the authors, "the data was translated by non-expert translators hired over Mechanical Turk and so it is of mixed quality." To our knowledge, this is the only publically available resource of parallel sentences extracted (though not automatically) from Wikipedia.

3 Wikipedia Data

Given the above amount of research dedicated to extracting parallel data from Wikipedia, it is rather unexpected that such publically available resources are virtually non-existent. The main reason for this absence is probably the high amount of computing resources (both in time and memory) necessary to run the proposed algorithms. This is why we have considered leveraging our previous work on this subject (Ştefănescu et al., 2012) and using LEXACC for harvesting parallel data from the entire Wikipedia. Our goal is to provide a large collection of parallel sentences to the research community for three language pairs: English-German, English-Romanian and English-Spanish. Three of the involved languages (i.e. English, German and Spanish) are listed among those having the top number of Wikipedia articles, while Romanian is also richly represented with almost 200,000 articles.

"Wikipedia is a free, collaboratively edited, and multilingual Internet encyclopedia supported by the non-profit Wikimedia Foundation." (cf. http://en.wikipedia.org/wiki/ Wikipedia). At the moment this paper was written (December 2012), it had over 4.1 million English articles containing embedded links to articles on the same subjects (see Fig. 1), but in different languages. According to Wikipedia, in December 2012 there were 285 languages for which it contained articles, making it the largest publically available collection of comparable documents.

Wikipedia articles can be downloaded by going to the URL[7] containing the so-called "database backup dumps". Wikipedia states that these dumps contain "a complete copy of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML". During May and June 2012, we downloaded the dumps having the label "current versions only" for English (e.g. file enwiki-20120601-pages-meta-

[7] http://dumps.wikimedia.org/backup-index.html

current.xml.bz2), German, Romanian and Spanish (see Table 1 for quantitative details). Parsing the English XML dump, we kept only the "proper" articles containing links to their corresponding articles in the other three languages. By proper articles we mean those that are not *talks* (e.g. Talk:Atlas Shrugged), *logs* (e.g. Wikipedia:Deletion log), *user related articles* (e.g. User:AnonymousCoward), *membership related articles* (e.g. Wikipedia:Building Wikipedia membership), *manuals* and *rules related articles*, etc.

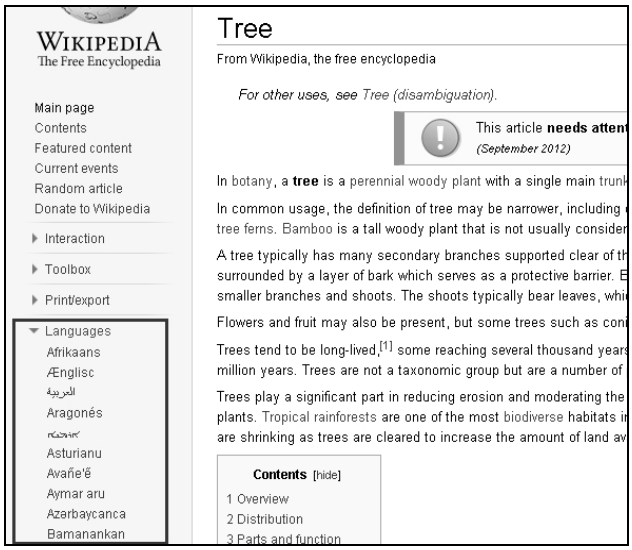


Fig. 1. Most Wikipedia articles contain links (lower-left corner in this image) to articles covering the same subject (e.g. Tree), but written in many different languages

Most of the proper articles do not contain colon characters within their titles, but not all of them (e.g. Blade Runner 3: Replicant Night). Each such article was processed using regular expressions to remove the XML mark-up in order to keep only the raw text, which was saved into a separate file. The non-textual entries like images or tables were also cleaned off. The articles identified as the corresponding articles in the other languages received the same treatment.

Table 1. Figures about the Wikipedia dumps, considering only proper articles

Language	English	German	Romanian	Spanish
Wiki dump date	June 01	May 08	June 10	June 11
Archive size on disk	15.4 Gb	3.9 Gb	277.9 Mb	2.0 Gb
Unpacked XML size	77.6 Gb	15.3 Gb	1.5 Gb	9.6 Gb
# Documents	3,975,895	1,406,603	180,234	894,378

We ended up with the lists of all comparable documents for all our language pairs and the documents themselves, containing only raw text. Since the Romanian Wikipedia had the fewest articles (180,234: 22 times less than the English one and almost 8 times less than the German one), the list of comparable documents for English – Romanian was also the shortest, containing almost 6 times less pairs than the English – German list (see Table 2).

Table 2. The number of comparable documents identified for each considered pair of languages

Language Pair	# Comparable Documents	Size on disk
English-German	715,555	2.8 Gb (English)
		2.3 Gb (German)
English-Romanian	122,532	778.1 Mb
		198.9 Mb
English-Spanish	573,771	2.5 Gb
		1.5 Gb

4 Extracting parallel sentences

With all the comparable documents in place, the next step was to employ LEXACC for harvesting parallel sentences. As input, this tool requires lists of source and target documents and, for a better accuracy, their correspondence. In order to ease its job, we first split the documents into sentences using a freely available sentence splitter[8] based on a Maximum Entropy classifier (Tufiş et al., 2008). This tool uses features that are language independent and though not as accurate as a language-aware sentence splitter, it achieves good results on most Indo-European languages. Furthermore, with the purpose of reducing LEXACC's running time, we partitioned the lists of mapped documents that had to be fed to it into smaller lists containing no more than 50,000 pairs. Consequently, we had 15 such lists for English-German, 3 for English-Romanian and 12 for English-Spanish. LEXACC was run for all these sub-lists in both directions (since its results are not symmetrical) for each language pair. We used as supplementary parameters (i) a flag signaling that the documents were already sentence split and (ii) that we want to keep all sentence pairs for which the assigned translation score was greater than 0.1. The tool was run on an x64 machine having an Intel Core I7 CPU @ 3.33 GHz and 16 GB of RAM. To run LEXACC for an English-German list (list.txt) of document pairs, one needs to use the following command line:

```
lexacc64.exe --docalign list.txt --source en --target de
--output results.txt --param seg=true --param t=0.1
```

LEXACC running time depends on both the size of the collection, but also on the size and quality of the dictionaries it uses as resources. This is why, even if the number of comparable documents was smaller, the running time for English-Romanian

[8] http://nlptools.racai.ro/nlptools/index.php?page=pwiki

was comparable to the others. Table 3 shows the running time needed by LEXACC to extract the parallel sentences. The English-Romanian dictionaries were extracted from the JRC-Acquis corpus and complemented with translation pairs from the Princeton WordNet (Fellbaum, 1998) to Romanian WordNet conceptual alignment (Tufiş et al., 2008). Every English word belonging to a synset is paired with all Romanian words in the corresponding synset and all inflectional variants of the two translation equivalents are also generated.

Table 3. LEXACC running time and the size of dictionaries used as resources

Language Direction	Running time		Dictionary size
	Minutes	Days	on disk
English-German	11,949	8.29	13.6 Mb
German-English	8,973	6.23	13.0 Mb
English-Romanian	8,583	5.96	283.4 Mb
Romanian-English	2,296	1.59	283.4 Mb
English-Spanish	9,786	6.79	15.0 Mb
Spanish-English	8,955	6.21	18.7 Mb
Total	50,542	35.07	657.1 Mb

We used the default LEXACC resources (dictionaries automatically extracted with GIZA++ (Och and Ney, 2003) from the JRC-Acquis). These resources were not available for English-Spanish and we also applied GIZA++ (with the standard parameterization) on both Europarl and JRC-Acquis parallel corpora to obtain them.

For the whole exercise, LEXACC's running time only exceeds one month, but still, this is a short time given the number of comparable documents to be analyzed.

Having parallel sentences in both directions for all language pairs, for each such pair we computed the union of the two sets of data (source-target and target-source), keeping the larger score for those sentence pairs appearing in both sets. This strategy is supported by the facts that LEXACC assigns high translation scores only if certain criteria for determining the translation cohesion are met, and, more often than not, the information needed in order to meet these criteria is not necessarily found in both directions. Moreover, LEXACC's translation similarity measure is tuned to achieve a better precision at the expense of recall and thus, keeping the maximum of the translation similarity score of a sentence pair discovered from both directions ensures the growth of the final parallel dataset. Finally, every sentence pair occurs only once within the entire merged collection (duplicates were eliminated).

All the sentence pairs we extracted are publically available and can be downloaded from http://ws.racai.ro:9191/repository/search/?q=Parallel+Wiki.

5 Statistics of the Extracted Parallel Corpora

The number of parallel sentences extracted by LEXACC is remarkable (see Tables 4 and 5). It is important to notice that the quantity of data acquired for English-Spanish is far greater than the others. This might be explained by the fact that Spanish articles

contain much more translations from English documents (or vice-versa). In total, we ended up with 429.7 Mb of data for English-German, 214.5 Mb for English-Romanian and 1.5 Gb for English-Spanish.

Table 4. Number of parallel sentences extracted for all language pairs at different thresholds

Score	English-German	English-Romanian	English-Spanish
0.9	38,390	42,201	91,630
0.8	119,480	112,341	576,179
0.7	190,135	142,512	1,219,866
0.6	255,128	169,662	1,579,692
0.5	322,011	201,263	1,838,794
0.4	412,608	252,203	2,102,025
0.3	559,235	317,238	2,656,915
0.2	929,956	449,640	3,850,782
0.1	1,279,166	683,223	5,025,786

Table 5. The total number of words (alpha-numeric tokens) in the parallel sentences extracted for all language pairs

Score	English-German		English-Romanian		English-Spanish	
	English	German	English	Romanian	English	Spanish
0.9	553,967	543,126	813,595	828,448	1,125,621	1,158,173
0.8	2,076,963	2,010,170	2,355,819	2,399,120	10,503,793	11,285,236
0.7	3,494,316	3,370,622	2,986,957	3,036,061	23,729,717	25,793,126
0.6	4,891,202	4,697,714	3,576,837	3,634,076	31,021,822	33,705,684
0.5	6,452,520	6,185,955	4,261,836	4,261,836	36,511,538	39,544,692
0.4	8,469,945	8,131,765	5,414,919	5,481,501	42,315,752	45,564,696
0.3	11,796,524	11,352,915	6,886,196	6,962,520	54,931,781	58,524,121
0.2	22,087,957	21,492,219	9,956,201	10,056,323	88,567,223	93,046,528
0.1	32,199,871	31,537,172	16,274,551	16,420,141	122,760,209	128,131,966

Table 6. The average / standard deviation for the number of words (alpha-numeric tokens) in sentences

Score	English-German		English-Romanian		English-Spanish	
	English	German	English	Romanian	English	Spanish
0.9	14.4 / 8.9	14.1 / 8.6	19.3 / 11.0	19.6 / 11.2	12.3 / 8.3	12.6 / 8.7
0.8	17.4 / 9.2	16.8 / 8.8	21.0 / 10.9	21.4 / 11.2	18.2 / 10.7	19.6 / 11.7
0.7	18.4 / 9.8	17.7 / 9.4	21.0 / 10.8	21.3 / 11.1	19.5 / 10.9	21.1 / 12.0
0.6	19.2 / 10.0	18.4 / 9.6	21.1 / 10.8	21.4 / 11.2	19.6 / 10.8	21.3 / 11.9
0.5	20.0 / 10.5	19.2 / 10.0	21.2 / 10.7	21.5 / 11.0	19.9 / 10.9	21.5 / 11.9
0.4	20.5 / 10.5	19.7 / 10.0	21.5 / 10.6	21.7 / 10.9	20.1 / 10.9	21.7 / 11.9
0.3	21.1 / 10.3	20.3 / 9.8	21.7 / 10.7	21.9 / 11.0	20.7 / 10.9	22.0 / 11.8
0.2	23.8 / 10.9	23.1 / 10.8	22.1 / 10.5	22.4 / 10.9	23.0 / 11.9	24.2 / 12.5
0.1	25.2 / 11.4	24.7 / 11.4	23.8 / 11.5	24.0 / 12.0	24.4 / 12.5	25.5 / 13.2

Looking at the average lengths of the sentences (see Table 6), one may notice that the higher the threshold, the shorter the sentence pairs LEXACC finds. This means that the application is more confident when assigning higher scores to shorter sentence pairs. The reader can also deduce that in general, expressing the same statement requires fewer words in German than in English and more words in Spanish than in English, while Romanian requires almost the same number of words as English.

Table 7. Examples of aligned sentences for English-Spanish having scores between 0.1 and 0.9

Score	English-Spanish sentence pair
0.9	The law provides for freedom of assembly and association, and the government generally respected these rights in practice. La ley provee de libertad de asamblea y asociaciones, el gobierno generalmente respeta estos derechos en práctica.
0.8	The rising Swedish exodus was caused by economic, political, and religious conditions affecting particularly the rural population. El creciente éxodo sueco fue causado por condiciones económicas, políticas y religiosas que afectaban particularmente a la población rural.
0.7	After she and her younger brother Andreas began to get successful in skiing - Hanni won the gold medal in slalom at the 1974 World Championships - the family was granted Liechtenstein citizenship. Después ella y su hermano Andreas llegaron a tener éxitos esquiando - Hanni llegó a ser la Campeona Mundial de Slalom en 1974 - a la familia se le concedió la ciudadanía de Liechtenstein.
0.6	Clairemont is a suburban neighborhood in northern San Diego. Clairemont es un barrio localizado en la ciudad de San Diego.
0.5	The origin of the name manganese is complex. El dióxido de manganeso se utiliza como cátodo.
0.4	Although the fossil record of pycnogonids is scant, it is clear that they once possessed a coelom, but it was later lost, and that the group is very old. Los fósiles conocidos de mayor edad son del Devónico, aunque dada su posición sistemática, el grupo debe ser mucho más antiguo.
0.3	Prayer vigils were also held on the second anniversary of the raid in Waterloo and Postville. La redada de Postville fue una redada en una planta empacadora de productos cárnicos en Postville, Iowa.
0.2	Although it sought to avoid entering the war, Spain did make plans for defence of the country. La situación de colaboración con los agentes del Eje, principalmente alemanes, en España era de conocimiento público.
0.1	Three years before events of the game, Dick disappeared from the Hamilton household and has not been seen since. Es asesinado por Dick Hamilton en un ataque de furia, al negar que Alyssa fuese criada como una "Rooder".

Looking at the standard deviations, we see high values which mean that the lengths of the sentences vary a lot and that the normal distributions of the lengths are flat.

The score LEXACC assigns to each extracted pair is an intrinsic measure of the degree of parallelism between the sentences in that pair. Manually analyzing the extracted data, we came to the conclusion that the pairs having scores above 0.4 can be easily considered comparable, while those above 0.6 can be considered parallel. It is up to the user to decide which threshold (s)he wants to use, depending on the personal view of the translation equivalency relation. This can be more rigid, meaning that the user is looking for word-to-word translations, or it can be more permissive, when the user is also satisfied with cross-lingual paraphrases. Table 7 shows English-Spanish sentence pairs with different scores. As the scores decrease, the sentence pairs are less and less reciprocal translations. In the same table, we can see that under the 0.6 value, the degree of parallelism decreases progressively. Yet, one can still find parallel pairs having low value scores assigned. This is probably because LEXACC did not have enough information to assign a higher score. For example, the following sentence pair received a score value of 0.288, which can be considered to be too low:

- The plans were revised just once after construction began, when certain technical difficulties arose.
- Tras el comienzo de las obras el plano aún tuvo que alterarse una vez, para sortear las dificultades técnicas que surgieron.

6 Conclusions

This paper describes a collection of parallel sentences extracted from Wikipedia for three pairs of languages: English-German, English-Romanian and English-Spanish. To do this, we employed LEXACC, a tool for extracting parallel sentences from comparable corpora, developed during the ACCURAT project. Each sentence pair is assigned a score which is a translation similarity measure for the sentences forming the pair. The entire collection of sentence pairs is publically available and can be downloaded from: http://ws.racai.ro:9191/repository/search/?q=Parallel+Wiki. It offers the scientific community almost 7 million comparable sentences, out of which more than 2 million can be safely considered parallel, having a translation similarity score above 0.6. Although several other researchers conducted experiments on extracting parallel sentences from Wikipedia for various language pairs, the volume of their data is, in general, much smaller than in our case. The reason is partly due to initial available Wikipedia data for the considered language pairs, but mainly due to the merits of LEXACC technology[9].

We can conclude that LEXACC is a robust tool which performs well on corpora having Wikipedia's level of comparability and that the whole experiment can be repeated for any other pair of languages in Wikipedia. Certainly, the most productive experiments would be run on well-covered language pairs. Moreover, given the large number of document pairs we had to consider (the entire Wikipedia for three pairs of

[9] http://nlptools.racai.ro/nlptools/index.php?page=lexacc ; see also http://ws.racai.ro:9191/

languages: over 10 Gb of raw text in comparable documents), LEXACC is an efficient tool as it acquired about 2.1 Gb of data (more than 20%) and also a fast tool as the entire running time took a few days over a month (on a fast computer).

A reasonable question that arises refers to the evaluation of the percent of data that can be extracted in this manner from a comparable text. We are convinced that this percent is highly dependent on the level of comparability of the input data. Since this aspect is very hard to be evaluated, the above question can possibly be answered only after conducting much more similar experiments. In the near future, we will evaluate the quality of the extracted data by using it for building translation models that will be tested using our SMT systems.

Acknowledgements

This work has been supported by the METANET4U SPS European project under the Grant Agreement no. 270893 and by the ACCURAT project within the FP7 Program under the Grant Agreement no. 248347.
The authors are also grateful to the anonymous reviewers for their valuable recommendations and comments, which helped them improve the quality of the paper.

References

1. Adafre, S. F., and de Rijke, M.: Finding similar sentences across multiple languages in wikipedia. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, pages 62-69 (2006).
2. Birch, L., Callison-Burch, C., Osborne, M., and Post, M.: The Indic multi-parallel corpus. http://homepages.inf.ed.ac.uk/miles/babel.html (2011).
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press (1998).
4. Gale, W. A., and Church, K. W.: A Program for Aligning Sentences in Bilingual Corpora. Computational Linguistics 19 (1), pages 75-102 (1993).
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the ACL Companion Volume Proceedings of the Demo and Poster Sessions, Prague, pages 177-180 (2007).
6. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit Vol. 5 (2005).
7. Mohammadi, M., and Ghasem-Aghaee, N.: Building bilingual parallel corpora based on Wikipedia. In Computer Engineering and Applications (ICCEA), 2010 Second International Conference on, Vol. 2, pages 264-268. IEEE (2010).
8. Och, F.J., and Ney, H.: A systematic comparison of various statistical alignment models. Computational linguistics 29.1, pages 19-51 (2003).
9. Quirk, C., Udupa, R., and Menezes, A.: Generative Models of Noisy Translations with Applications to Parallel Fragment Extraction. In: Proceedings of the MT Summit XI, European Association for Machine Translation, pages 377-384 (2007).

10. Rauf, S., and Schwenk, H.: Parallel sentence generation from comparable corpora for improved SMT. Machine Translation, 25(4), pages 341-375 (2011).
11. Resnik, P., and Smith, N. A.: The web as a parallel corpus. Computational Linguistics, 29(3), pages 349-380 (2003).
12. Smith, J. R., Quirk, C., and Toutanova, K.: Extracting parallel sentences from comparable corpora using document level alignment. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 403-411. Association for Computational Linguistics (2010).
13. Ştefănescu, D., Ion, R., and Hunsicker, S.: Hybrid parallel sentence mining from comparable corpora. In Proceedings of the 16th Conference of the EAMT, Trento, Italy (2012).
14. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D. and Varga, D.: The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pages 2142-2147, Genoa, Italy. ELRA. ISBN 2-9517408-2-4 (2006).
15. Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
16. Tillmann, C.: A Beam-Search extraction algorithm for comparable data. In Proceedings of ACL, pages 225-228 (2009).
17. Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A. and Ştefănescu, D.: Romanian Wordnet: Current State, New Applications and Prospects. In Proceedings of 4th Global WordNet Conference, GWC-2008, pages 441-452, Szeged, Hungary. University of Szeged, Hungary. ISBN 978-963-482-854-9 (2008).
18. Tufiş, D., Ion, R., Ceauşu, A., Ştefănescu, D.: RACAI's Linguistic Web Services. In Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008, Marrakech, Morocco (2008).
19. Wu, D.: Aligning a parallel English-Chinese corpus statistically with lexical criteria. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 80-87. Association for Computational Linguistics (1994).
20. Yasuda, K., and Sumita, E.: Method for building sentence-aligned corpus from wikipedia. In 2008 AAAI Workshop on Wikipedia and Artificial Intelligence (WikiAI08). (2008).
21. Zhao, B., and Vogel, S.: Adaptive parallel sentences mining from web bilingual news collection. In Proceedings of the 2002 IEEE International Conference on Data Mining, page 745. IEEE Computer Society (2002).

Improving term extraction with linguistic analysis in the biomedical domain

Wiktoria Golik[1], Robert Bossy[1], Zorana Ratkovic[1,2] and Claire Nédellec[1]

[1]MIG INRA UR1077, Domaine de Vilvert, F-850 Jouy-en-Josas, France 2
[2]LaTTiCe UMR 8094 CNRS, 1 rue Maurice Arnoux, F-92120 Montrouge, France
Wiktoria.Golik@jouy.inra.fr, Robert.Bossy@jouy.inra.fr, Zorana.Ratkovic@jouy.inra.fr, Claire.Nedellec@jouy.inra.fr

Abstract.
This paper presents a linguistic-based approach to term extraction from corpora in the biomedical domain. The method is based on an analysis of terms and their context that verify linguistic constraints. It focuses on participles and prepositional complements. The purpose of our approach is to obtain terms that are relevant for knowledge acquisition applications, such as the creation and enrichment of terminologies and ontologies. We report on the evaluations we conducted by applying two complementary strategies, using a reference terminology and a manual validation. They were applied to two corpora of differing genres and Life Science domains, namely pharmacology patents and animal physiology scientific articles. Our work shows that the linguistic analysis-based developments significantly improve the extraction results. The method is especially efficient when dealing with gerunds and *to* prepositional modifiers.

Keywords: term extraction, biomedical corpora, linguistic approach

1 Introduction

The amount of biomedical information is growing exponentially. Most of this information is made available through domain literature and is expressed in natural language (Jensen *et al.*, 2006). The need to automatically process this large amount of data has led to advancement in the field of biomedical text mining in the past several years. Most of the work has focused on Information Extraction (IE). The use of terminological resources and ontologies has been found to be necessary for high-quality IE (Nenadic *et al.*, 2006; Bodenreider, 2006). Recent developments in NLP and term extraction methods offer a powerful and efficient way to design terminological resources and facilitate access to scientific information.

In recent years, several approaches have been proposed for the acquisition of terms from text. Reviews by (Pazienza *et al.*, 2005; Zhang *et al.*, 2008) have described and compared the most popular techniques. Traditionally, term extraction methods are divided into linguistic, statistical and hybrid ones. Linguistic methods attempt to identify terms by their linguistic properties, while the statistical methods are based on frequency, association and the distribution of terms in documents. By handling the particularities of language structure, linguistic tools usually provide a large number of

well-formed and diverse candidate terms (CTs). Conversely, statistical approaches are knowledge-poor, but are fully automatic and are language and domain independent. Also, they provide noisier results due to their reliance on the frequency and length of CTs. Hybrid systems combine the advantages of both approaches, using linguistic techniques for term acquisition and statistics for term ranking (Sclano & Velardi, 2007; Wang *et al.*, 2007).

In this paper, we focus on the linguistic approach as a critical step for term extraction. The paper describes our method, which aims to improve the quality of extracted terms from biomedical corpora. The purpose is to obtain relevant terms for knowledge acquisition applications, such as the building of terminologies and ontologies. Unlike other applications (*e.g.* document ranking), the well-formedness of terms is crucial. Our method is based on the use of appropriate filtering and the processing of important linguistic structures that are commonly found in biomedical texts and beyond. In particular, they include structures containing prepositional noun phrases (NN PREP NN) and participles (past participles and gerunds). The paper details two different evaluations and discusses their complementarities.

The term extraction experiments were performed on two different types of corpora from two biomedical subdomains: patents and scientific papers from the pharmacology domain, and scientific journals from the animal physiology domain. In the next section we describe previous work in this field. Section 3 details the motivation of our work. The method is described in Section 4. Section 5 details the experiments. The results are reported in section 6, while in section 7 we discuss the results and conclude.

2 Related Work

Statistical methods have been found to be very effective at ranking CTs (Zhang *et al.*, 2010) and research in this field has been very active in recent years. Linguistic-based methods are founded on a deep analysis of different linguistic phenomena that can be observed in the data. Traditionally, most of the linguistic and hybrid approaches focus on noun phrase (NP) extraction, since NPs usually contain domain relevant semantic information (Justeson and Katz, 1995). The extraction generally targets multi-word units, since it has been found that 85% of domain-specific terms are multi-words (Nakagawa & Mori, 2002).

A typical extraction process includes POS tagging, tokenization, chunking and the use of linguistic patterns (Sclano & Velardi, 2007, Wermter & Hahn, 2005). POS tagging has a strong impact on the extraction of CTs, most notably on chunking, which is based on POS tags. POS tagging is particularly important for the correct extraction of phrases containing participles, since participles can play the role of verbs, adjectives or nouns. Despite the high performance of existing taggers, some errors do occur. The problem of participle tagging has been addressed in tagging annotation guidelines (Santorini, 1990) and syntactic parsing (Hara, Miyao & Tsuji, 2009). However, it still remains problematic for term extraction systems whose results strongly depend on the quality of the tagging.

The use of well-defined linguistic patterns improves the extraction accuracy and correctness. As described in (Frantzi, Ananiadou & Mina, 2000), they are built in order to extract frequent and simple structures (NN NN) as well as more complex ones, such as those including prepositions. The more a pattern is complex, the more variable and unpredictable the results may be, including incorrect terms. Simple patterns are less error prone, but they are too restrictive and less productive. Some systems use syntax-based endogenous disambiguation to handle noisy data. For example, Syntex (Bourigault, 2007) learns and compares syntactic contexts of CTs in order to distinguish relevant and irrelevant forms. This comparison results in a better extraction of prepositional phrases that typically depend on the context.

The quality of extraction also depends on the capability to extract various forms of terms. Variant extractors, such as FastR (Jaquemin, 1999) improve the completeness of extracted NPs. A large number of variants, including prepositional ones (*e.g.* NN at NN, NN in NN) are identified using different meta-rules (*i.e.* permutation, insertion, coordination). The abundance and accuracy of the produced variants depend on terms that are already recognized and accepted as valid (attested terms), as well as certified resources that are used as a starting point.

The use of existing domain-specific resources is another valuable way to enhance term extraction. POS-tagged resources help to deal with the POS tagging quality problem, as well as the lack of extraction patterns (Aubin & Hamon, 2006; Roberts et al., 2008).

Finally, the quality of CTs depends on their well-formedness and domain relevance. As already mentioned, this task is often handled by statistical approaches or the use of limited but efficient techniques such as stop lists or simple linguistic filters. TermExtractor (Sclano & Velardi, 2007) is a hybrid tool that relies on both. Some systems additionally use contextual information to capture the domain relevance of terms. In (Frantzi et al. 2000) the linguistic and statistical analysis of nested terms serves to identify domain-specific term markers. More recently, a similar method has also been applied to filter out incomplete phrases (Gojun *et al.*, 2012).

3 Motivation

There are many resources that have been developed for the biomedical domain, such as the UMLS metathesaurus (Bodenreider, 2004) and Gene Ontology (Harris *et al.*, 2004). Due to the fast evolution of the field there is a constant need to produce new resources and to complete existing ones. Data from different biomedical subdomains cannot be semantically processed using a single resource. The diversity of the goals further stresses the importance of having different and new resources for different datasets and different tasks. There is a need for efficient and automatic systems that can be used to create and guide the creation of such resources (Nédellec *et al.*, 2010).

Our work focuses on the improvement of term extraction of biomedical corpora that produces grammatically well-formed and application relevant terms. Most term extractors focus on NPs without taking into account possible prepositional phrases. This is due to the high attachment ambiguity. However, NPs with prepositional modi-

fiers such as *NN in NN*, *NN for NN*, *NN at NN* and *NN to NN* are interesting and useful to consider. The exploration of biomedical corpora, as well as the consultation of domain experts, have led us to conclude that these structures can generate highly relevant domain terms. They can considerably enrich specific domain resources where prepositional phrases are infrequent.

Next, we consider that past and present participles, when correctly POS tagged and extracted, can improve the extraction comprehensiveness, in particular in the biomedical domain where they are frequently employed.

We also observed that the extracted terms often contain terms that are referential or too general. These terms are irrelevant to the application domain and should be filtered out. We propose to improve term extraction in three steps: (i) extension of extracted terms by considering prepositional phrases (ii) supervision of the extraction through an enhanced processing of *–ing* and *–ed* forms (iii) elimination of irrelevant terms through the use of filters.

4 Methods

Our method aims to improve term extraction of biomedical corpora, for the creation of lexical resources, such as terminologies or ontologies. The method is based on a linguistic analysis of biomedical texts. We make no assumptions as to the nature and exact usage of the resource. Rather, we aim to extract terms which are well-formed, complete from a syntactic and semantic point of view and could be useful for a given application. We leave the term relevance decision to experts. Our approach captures the linguistic phenomena that are found in biomedical corpora differing in style and genre. For this reason it will be broadly applicable. We did not use a machine learning (ML) approach since this requires annotated data, which is not available and is costly to produce. Moreover, ML approaches always introduce bias with respect to the training corpus that is used. The extraction is based on linguistics patterns reinforced by additional context-based rules in order to handle specific prepositional phrases and participles. We are unaware of any previous work that combines both in order to enhance the extraction of such structures.

4.1 Extraction of Structures Containing Prepositions

Prepositional attachment resolution is a well-known problem in NLP, especially for syntactic parsing (Ratnaparkhi *et al.*, 1994; Nakov & Hearst, 2005). Due to their high degree of ambiguity, PPs are often not taken into account in term extraction. In the case of shallow parsing the prepositions are usually treated as boundaries of chunks (Ash & Daelemans, 2009), except for the frequent prepositional structure *NN of NN*. PP ambiguity is strongly related to the nature of predicate-argument structures, and more precisely to the difference between an argument and an adjunct (Grimshaw, 1992). According to the context, PPs act as arguments or adjuncts. This ambiguity is difficult to resolve automatically since it depends on both syntax and semantics. Our corpus analysis revealed terms containing PPs, which were both well-formed and

domain relevant. A closer look at the most frequent prepositions showed that most frequently they are either arguments of verbs (such as *by* and *in*) or are adjuncts and parts of NPs (such as *of*, *to* and *at*). Our work focuses on the latter case. Since the *of* preposition is treated in previous works, we focus on *at* and *to*. For instance, NPs with the preposition *at* often contain information about level, condition or period (*e.g.*, age at parturition, body weight at birth). NPs containing *to* denote reactions to different stimuli and situations (*e.g.*, susceptibility to mastitis, response to fish oil supplementation).

The extraction of terms containing *at* and *to* is done in two steps: (i) the application of extraction patterns that include the prepositions (*e.g.*, NN *to* NN or NN *at* JJ NN) (ii) the filtering of irrelevant attachments by a set of five context based rules (see Table 1). Their role is either to trigger the extraction of relevant PPs or to prevent the extraction of irrelevant ones. For instance, for CTs containing *to*, the first rule in the table checks if the structure NN *to* NN is preceded by *from* or *by* (*e.g.*, from mother to young), in which case the CT is not extracted because *to* is directly related to *from* and not to the NP. The proposed rules are generic in order to be applicable to different corpora.

Table 1. Context-based rules for the extraction of *to* and *at*.

Context-based rules	**Relevant POS tag**
[from|by][not SENT][**to**]	Reject
[not NN] [not V][**to**]	Reject
[not V|VVN][**to**]	Reject
[NN|VVN and not stop-list][**to**]	Reject
[stop-list][not SENT][**at**] e.g. *weight at birth*	AT

4.2 Candidate Term Refinement by Filtering

Filtering means the automatic removal of forms considered to be non-terms, similarly to Pazienza *et al.* (2005). We use basic linguistic filters that are efficient and easy to build and maintain. The method aims to improve the extraction results by filtering out two kinds of irrelevant terms. First, terms which are structurally incoherent (*i.e.* invalid). The filter is simple, yet very efficient. Second, terms which are structurally coherent, but are referential or too vague (*i.e.* semantically poor). The aim of the filter is to remove terms that are grammatically well-formed, but that are not useful for any domain application. Most often, they are referential expressions where the context is needed to interpret the term, or terms that reflect the writing style and do not convey domain knowledge.

Filtering of Invalid Forms

The quality of the extraction depends on the quality of preceding tokenization and POS tagging steps. The filtering handles incorrectly tokenized or POS-tagged CTs. The filters capture: surface forms that start or end with invalid characters (*e.g.*, +, ~, *, \, .); surface forms that start with coordination marks (*e.g.*, and, or), contain only parenthesis or square bracket (*e.g.*, B2(lipid source), start or end with a unit of measurement (*e.g.*, kg clozapine, 9 mm), or contain only numbers (*e.g.*, 1666-1673).

Filtering is also used to handle function words that are traditionally filtered out due to their high frequency and their lack of semantic information, such as definite and indefinite articles, demonstratives and *wh*-determiners (*e.g.*, the, these, which, each).

Filtering of Semantically Poor Terms

The filtering performs a preliminary semantic refinement of CTs. It identifies correctly extracted NPs that cannot be considered as true terms from a semantic point of view. They can be divided into four main types:

- NPs usually containing non discriminative modifiers (*e.g.*, important, particular, useful, various, certain, amount of)
- NPs that depend on the context in order to be properly interpreted (*e.g.*, day 33, position 1978*)*; they often include comparatives (*e.g.*, greater DMI, higher number of assays)
- NPs directly linked to the nature and style of the corpora (*e.g.*, embodiment, point of view, above-mentioned feature, present experiment)
- Named entities related to the references present in the documents, dates (*e.g.*, Smith et al., November 1986)

4.3 The Extraction of Gerunds and Past Participles

The last part of our method focuses on the POS-tagging of participles, which is a common problem in NLP, notably in the biomedical domain (Teteisi & Tsuji, 2006). According to the context, the participles play either the role of verbs, adjectives or nouns (*i.e.*, binding). These three POS tags are particularly difficult to distinguish and the context is usually discriminant. The erroneous tagging of participles usually leads to the omission of relevant NPs. The number of NPs with *–ed* and *-ing* is high in most biomedical corpora. While there are POS taggers that have been adapted to the biomedical domain, the problem still persists. We do not consider retraining a tagger for two main reasons. First, such a process requires manually annotated data. Secondly, the tagger will be influenced by the corpora that it is trained on.

To improve the completeness of the extraction, we propose to supervise the tagging stage using five context-based rules. These rules take into account the words surrounding an *–ing* or *–ed* form and their POS tags. Additionally, the rules use a stop list of forms that are always verbs (*i.e.* being, using, getting). The list was collected from the corpus and tested using the criterion proposed in (Santorini, 1999). For instance, an *–ing* form (not in the stop list) preceded by *of* and not followed by a verb or punctuation mark will be tagged as NN (*e.g.* day of calving, role of farming).

Table 2. Context-based rules for the disambiguation of participle POS tags. [ing*= ing and not stop-list]

Rule	Relevant POS tag
[DT|JJ|SENT] [–**ing***)][NN|NNS|NP|,] e.g. *eating quality*; *a training period*	NN
[DT|JJ][–**ing***][JJ] e.g. *increasing perinatal mortality*	JJ
[of][-***ing****][not V|,|SENT] e.g. day *of calving* ; role *of farming*	NN
[of][-***ing****][DT|JJ|PP|WDT] e.g. accuracy *of predicting the* percentage	VVG
[-***ed***][NN|NP|JJ|NNS] e.g. autumn *saved* pasture; *immunized animals*	JJ

5 Experiments

5.1 The Dataset

Biomedical literature is very rich and diverse. The variation in document language significantly varies with respect to the scientific field and the document genre (Lippincott *et al.*, 2010). In our experiments, we use two corpora of different genres: patents and scientific papers. They also belong to two very distinct biomedical subdomains: pharmacology and animal physiology. They are representative of the heterogeneity found in biomedical texts. The patents belong to the legal literature and are characterized by a highly controlled structure and vocabulary, while the scientific papers express different structural constraints and language. Further, scientific papers describe experimental hypotheses, procedures and results. By dealing with these very different corpora, we demonstrate our method to be general and applicable to different biomedical domain texts.

Pharmacology Domain Corpora

For the first evaluation experiment we used the pharmacology corpora used for the term extraction challenge of the *Quaero*[1] project organized in 2010 and 2011 in which we participated (Mondary *et al.*, 2012). It consists of four corpora: three are made up of patents (C1, C2, and C3) and one of scientific papers (CA). The patents are the European patents from the A61K class of the ECLA classification on preparations for medical, dental or toilet purposes. For the evaluation, we reused the largest patent corpus (C3) and CA. C3 contains 157 patents (2,500,000 words). CA is made up of

[1] http://www.quaero.org/modules/movie/scenes/home/index.php?FUSEBOX_LANG=2

7,030 scientific paper abstracts (1,500,000 words) from the PASCAL database[2]. The extraction results were evaluated against a reference terminology (see section 5.3).

Animal Physiology Domain Corpus

The animal physiology corpus is made up of full-text papers from the *Animal* journal (Cambridge University Press) published until 2011. The corpus contains 697 scientific communication papers that cover a large number of subjects from the animal physiology domain. Scientific papers use descriptive language and are characterized by a high variability of expressions and by the frequent presence of specific linguistic forms such as gerunds. In this corpus these forms are used to describe the states or activities of animals (*i.e.* abnormal calving, adequate laying space, grazing behavior).

5.2 BioYaTeA

For our experiments we used the BioYaTeA[3] term extractor, an extended version of YaTeA (Aubin & Hamon, 2006). YaTeA's extraction method includes the detection of morpho-syntactic boundaries and the matching of parsing patterns. It also comprises exogenous (supervised) and endogenous (unsupervised) disambiguation. For the experiment, we used YaTeA version v.0.6.

We extended YaTeA with new syntactic patterns, context-based rules and the post-processing filtering described in section 4. This new version is called BioYaTeA. BioYaTeA is integrated into a generic NLP platform developed at INRA-MIG, namely the AlvisNLP pipeline (Nédellec *et al.*, 2008). It takes as input the results of the AlvisNLP tokenizer and the POS tagger TreeTagger (Schmid, 1994). BioYaTeA was designed to build domain specific ontologies for automatic fine-grained indexing of biomedical texts (Nédellec *et al.*, 2010) for semantic search engine applications. Here, we use it to measure the added-value of our method. However, our approach is universal and it could be implemented with any term extractor, whether it is a statistical, linguistic or hybrid one.

5.3 Evaluation Methods

The evaluation of extracted CTs is a difficult and costly task. It requires the definition of what a term is and how to determine it (Pazienza, 2005; Vivaldi, 2007). Early works aimed to define termhood (Kageura & Umino, 1996) and the different ways of measuring it (Ananiadou *et al.*, 1998). Difficulties remain due to the complex nature of terms and the lack of a general consensus. Also, the evaluation of corpus term extraction results depends on the target application, the strategy being used and the nature of the corpora (Zhang *et al.*, 2008).

Traditionally, the results are evaluated using (i) a comparison to a reference terminology (ii) expert judgment with respect to a target end-user application (*i.e.* infor-

[2] The multidisciplinary bibliographical database produced by INIST-CNRS.PASCAL can be found on INIST's official website: http://inist.fr/spip.php?article11

[3] BioYaTeA is available at http://search.cpan.org/~bibliome/Lingua-BioYaTeA/

mation extraction). The advantage of the reference-based approach is that it is fast, fully automatic, and it allows for the use of standard metrics such as recall, precision and F-measure. However, relevant corpus terms might be counted as irrelevant because they are missing from the reference. To be useful, the recall should be measured with respect to what is effectively extractable from the corpus. This requires a manual annotation of corpus terms. The reference-based approach is not an absolute measure, but is used to measure the relative performance of different tools.

Manual validation is time consuming and requires the participation of domain experts or knowledge engineers. It relies on human judgment that can vary from one person to another. However, this type of evaluation is the most efficient and popular since it evaluates the extraction quality for a targeted purpose.

Our experiments include both types of evaluation. They are fundamentally different in their goal; the reference-based evaluation allows us to check the domain relevance of CTs, while the manual validation aims to estimate the value of CTs according to the ontology building task.

Automatic Evaluation of Pharmacology Term Candidates

In order to evaluate the CTs extracted from the pharmacology corpora, we used the same evaluation method as proposed for the *Quaero* term extraction challenge (Mondary *et al.*, 2012). The aim of the original evaluation was to compare the results of different term extractors using pharmacology domain corpora. The extractions were compared to a gold standard reference terminology containing 76,466 terms. According to the protocol proposed in (Nazarenko & Zargayouna, 2009) the standard precision and recall metrics were adapted to terminological result evaluations by taking into account partial matches. The results of the *Quaero* evaluation showed that filtering played an essential role in the quality of the results. We repeated the same experiments in order to better characterize the impact of each improvement described in Section 4. We applied the same protocol with the same corpora, as well as the same gold reference (see section 6.1).

Manual Evaluation of Animal Physiology Candidate Terms

The second evaluation was done using the animal physiology domain corpus. The evaluation was manual and involved ten annotators who were not animal physiology experts, but were familiar with the biomedical domain and with the development of ontology driven IR and IE applications. To validate terms, they referred to detailed guidelines[4] as defined by a knowledge engineer.

A sample of terms was judged according to their correctness in general, as well as according to their potential relevance for the design of an ontology. The annotators pointed out CTs that were too general or rhetorical. This allowed us to measure the impact of the extraction improvements. Since the annotators relied on the guidelines and not their expertise, this suggests that their validation should be more consistent compared to a domain expert validation. The evaluation was performed using the TyDI interface (Nédellec *et al.*, 2010) that gives access to the context of the terms,

[4] http://bibliome.jouy.inra.fr/GuideEvaluateur.pdf

which can be critical in some cases. The validation of terms was double-blind, with two annotators assigned to each term. We measured the Cohen kappa inter-annotator agreement (Cohen, 1960).

6 Results

The evaluation compares three versions of the term extractor, from the basic to the most enhanced version: YaTeA, YaTeA with filter and BioYaTeA (*i.e.* YaTeA with filter and rules). BioYaTeA does not perform any additional processing of single-word CTs. However, we decided to include single-word CTs in the reference-based evaluation due to their presence in the reference terminology. For the manual validation we only kept multi-word CTs.

6.1 Pharmacology Corpora Extraction: Evaluation and Result Analysis

The three term extractors were applied to the two pharmacology corpora (C3 and CA). The extracted CTs were evaluated and compared to the reference terminology. The terminological precision (t-precision), the terminological recall (t-recall) and the terminological F-measure (t-F-measure) were calculated for each corpus and each extraction as displayed in Table 3.

Corpus		YaTeA	YaTeA+filter	BioYaTeA (YaTeA+filter+rules)
C3	TP	34.2	48.0 (+13.8)	52.9 (+4.9)
	TR	33.1	29.4 (-3.7)	29.1 (-0.3)
	TF	33.7	36.4 (+2.7)	37.5 (+1.1)
CA	TP	46.2	56.7 (+10.5)	55.5 (-1.2)
	TR	37.3	33.9 (-3.4)	33.9 (+0)
	TF	41.3	42.4 (+1.1)	42.1 (-0.3)

Table 3**.** Pharmacology corpora extraction evaluation results.

There are several general trends to be observed from Table 3. BioYaTeA increases the precision, while decreasing the recall. However, there is an overall improvement, as shown by the increase in F-measure. For both of the corpora, the filtering results in a global improvement. These results show that the filtering is efficient for both corpora, even thought they differ in size and genre. In particular, the filters avoided an overly greedy extraction, making the extraction more precise by keeping a large number of true positives.

The effect of the rules is not the same for the two corpora. Overall, the effect of the rules on C3 results is positive, as shown by the last column in Table 3. This is mainly due to an increase in precision. Surprisingly, the effect of the rules on the CA corpus is less pronounced. Their impact on the t-recall is negligible while the t-precision slightly decreased. Given that the rules target only specific structures (participles and prepositional NPs) they had a less significant impact on the results than the filters.

The analysis of the effects of the rules is complex because of possible interactions between rules and filters. Our hypothesis is that it may be due to a different coverage of the targeted syntactic structures. For instance, the rules that limit the extraction of unwanted prepositional and gerund NPs are responsible for the t-precision increase in C3. The more permissive rules handling the participial NP extraction may decrease the t-recall in CA, where these structures are particularly abundant.

6.2 Animal Corpus Extraction: Evaluation and Result Analysis

We performed two manual validations on the *Animal* corpus in order to evaluate the filtering and the rules.

Validation of the Filter

The extractor comparison sample was built using the two sets of CTs extracted by YaTeA: CTs rejected by the filter (within BioYaTeA) and CTs accepted by the filter. The manual evaluation focused on a subset of 1,125 candidate terms (0.5% of the total) that were randomly selected from the two CT sets. The proportion of CTs from each sample as related to the total had been kept during the sample selection. Each term was validated by two annotators in a double blind-mode and each annotator validated 225 terms. The annotators did not know which terms were rejected or accepted by the filter. The Cohen kappa inter-validator agreement was 0.92, which indicates a very high agreement. The results are displayed in Table 4. Most of the CTs retained by the filters were manually validated as correct by the human annotators (68%). Similarly, most of the terms rejected were validated as incorrect (77%). As with the pharmacology evaluation, the filter significantly improved the extraction result. A detailed analysis of CTs that were wrongly omitted by the filter led us to observe that 30% of them were incomplete forms, due to POS tagging or attachment errors. The remaining 70% were well-formed according to the annotators but were either irrelevant to the target application, or too general for a domain ontology (*e.g.*, models of work, subset of data, progressive increase), which is outside the scope of the term extractor.

Table 4. Validation of terms rejected or accepted by the filter.

	TOTAL (1125)	Rejected (124)	Passed (1001)
Correct terms	713 (63%)	27 (22%)	686 (68%, +5)
Incorrect terms	376 (33%)	95 (77%)	281 (28%, -5)
Controversial terms	36 (3%)	2 (1.6%)	34 (4%)

Finally, the analysis of false negatives showed that a large number of them were filtered out because they end with numbers (*e.g.*, FIL 2001) or start with a single capital letter. It is not within a general-purpose term extractor's ability to distinguish forms such as *N basis* (correct) from *S lambs* (incorrect). This suggests that the filter should

be enriched in order to take into account additional semantic domain-specific knowledge, when possible.

Validation of the Rules

In order to evaluate the effect of the context-based rules, we manually validated two sets of candidate terms: those extracted by YaTeA and those extracted by BioYaTeA. The filter was applied to both extractions. As with the previous validation we selected a random sample. We took care to retain the relative frequency of the syntactic structure distributions (Table 5).

Table 5. Number of terms extracted by two term extractors.

Type of structure	YaTeA + filter	BioYaTeA
-ing	21	81
-ed	15	324
to	41	12
at	0	17
Other	455	159
Total sampled	532	593
Total extracted	27589	31206

The prepositional structures with *at* were not extracted by YaTeA, but only by BioYaTeA. The number of expressions containing the *to* preposition was lower in the second extraction because of more restrictive rules. Conversely, the number of *–ing* and *–ed* forms was higher in the second extraction. This was due to the rules being more inclusive.

The annotators validated the same number of terms as for the previous validation and they referred to the same guidelines. The Cohen kappa was 0.78, which indicates a reasonably high agreement. The results in Table 6 show that for the specific syntactic structures that were tackled, the rules improve the results. In particular, for the *–ing* and *to* structures there is an increase in the number of correct CTs that were extracted and a decrease in the extraction of incorrect CTs. For the *at* CTs, the rules permit the extraction of such structures, which for the most part are correct (76%). The results for the *-ed* structures are less pronounced. There is an increase in both the number of correct and incorrect CTs. This suggests that the context-based rules concerning the *-ed* forms were too permissive. After a deeper analysis of the *-ed* false positives, we noticed that 38% of the CTs were validated as incorrect because they played the role of verbs, 45% were viewed as irrelevant for the domain (*e.g.*, detailed description of the dissection, improved likelihood, tested oils), 8% were incomplete or incorrectly extracted and 10% were validated as incorrect but no reason was specified.

The results confirm that the extraction of *–ed* NPs is a complex task. The extraction should be improved using an exhaustive stop list. Furthermore, an additional

deeper analysis of the syntactic context should be completed in order to better define the context of correct terms.

Table 6**.** Validation results of the terms extracted with and without rules.

	Correct		**Incorrect**		**Conflicts**	
	BioYaTeA	BioYaTeA + rules	BioYaTeA	BioYaTeA + rules	BioYaTeA	BioYaTeA + rules
-ed	53%	55%	20%	36%	27%	8%
-ing	48%	62%	38%	30%	14%	9%
to	41%	83%	51%	8%	7%	8%
at	-	76%	-	24%	0	0
Other	55%	50%	32%	41%	13%	8%
Total	53%	56%	34%	36%	13%	8%

Finally, the quality of the extraction of *other* CTs is surprisingly lower when using the rules. We noticed that most of these CTs (46%), despite their valid form, were incorrect because they were irrelevant as ontology concepts. The rest of the rejected NPs were incomplete or they were erroneously extracted. This shows that although the new context-based rules cannot capture semantic information, they are efficient at dealing with syntactic criteria. We need to extend the method by taking into account the semantic information, together with the syntactic information that the filters and rules handle.

7 Conclusion

Both experiments show promising and interesting results. First, the positive impact of the filtering is clear in both of the evaluations. The role of the context-based rules turned out to be more difficult to assess, especially using the gold standard approach. However, the more detailed analysis of the manual validation showed powerful rules (*to*, *at*, *-ing*), as well as concerns that need to be addressed more thoroughly (*–ed*). Finally, the results confirm that the relevance of CTs for a given application is an important criterion. Even thought well-formed, a part of the CTs were evaluated as incorrect because they were inappropriate for the application. To better address this problem, we plan on adding complementary techniques that examine the semantics of the CTs, such as distributional analysis (Harris, 1954) in order to group CTs into semantic clusters. Distributional semantics could bring us new relevant extraction patterns and help us to built appropriate stop lists. Building well-limited semantic clusters could be an efficient way to better distinguish domain and application relevant CTs.

In this paper, we presented the improvements designed to increase the quality, completeness and accuracy of extracted terms from biomedical corpora, with respect to the design of domain resources. They consist of linguistic based filtering of unwanted candidate terms and a rule-controlled extraction of NPs containing prepositions and participles. Two different evaluation strategies were applied, confirming the positive impact of such improvements for two different biomedical corpora. Further work should be conducted to confirm the results in other domains.

Acknowledgements

The authors thank the following persons for having validated the terms: Philippe Bessières, Tristan Bitard-Feildel, Paul Bui-Quang, Julien Jourde, Dialekti Valsamou and Pierre Warnier. This work was partially supported by the Quareo Programme funded by OSEO (French agency for innovation).

References

1. Aubin S. and Hamon T. (2006). Improving Term Extraction with Terminological Resources, in T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala (eds.), *Proc. of the Advances in Natural Language Processing*, FinTAL'06, LNAI 4139, Springer, p. 380-387, 2006.
2. Bodenreider, O. (2006) Lexical, terminological and ontological resources for biological text mining, in Ananiadou S., McNaught J., (eds.) *Text mining for biology and biomedicine*, Artech House, p. 43-66.
3. Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Researcher*, 32: supplement 1.
4. Bossy, R., Kotoujansky, A., et al. (2008). *Close Integration of ML and NLP Tools in BioAlvis for Semantic Search in Bacteriology.* In: Burger, A. et al. (eds.) Proceedings of the Workshop on Semantic Web Applications and Tools for Life Sciences. UK (2008).
5. Bourigault, D. (2007). Un analyseur syntaxique opérationnel: SYNTEX. *Mémoire d'Habilitation*, Université de Toulouse-le-Mirail.
6. Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
7. Frantzi, K., Ananiadou, S. and Mima, H. (2000). Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal of Digital Libraries*, 3(2):117–132.
8. Frantzi, K. T., Ananiadou, S., and Tsujii, J. (1998). The C-value/NC-value Method of Automatic Recognition for Multi-word Terms. In Goos G., Hartmanis J. & van Leeuwen J. (eds.), *Research and Advanced Technology for Digital Libraries: Proceedings of the Second European Conference*, ECDL'98 (Vol. 1513, p. 585–604). Lecture Notes in Computer Science. Berlin/ Heidelberg: Springer.
9. Gojun, A., Heid, U., Weissbach, B., Loth, C. and Mingers, I. (2012). *Adapting and evaluating a generic term extraction tool.* Proceeding of LREC-8.
10. Grimshaw, J. (1992). *Argument structure.* MIT Press, Cambridge, MA.
11. Harris, MA., et al. (2004). The Gene Ontology (GO) database and informatics resource. Nucleic Acid Research, 32 (Database issue).
12. Harris, Z. (1954). Distributional structure. *Word*, 10: 146-162.

13. Jacquemin C. (1999). Syntagmatic and paradigmatic representations of term variation. In *Proceedings of ACL'99*, p. 341-348.
14. Jensen, L. J., Saric, J. and Bork, P. (2006). Literature mining for the biologist: from information retrieval to biological discovery. *Nature Reviews Genetics*, 7: 119-129.
15. Justeson, J. S. and Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.
16. Kageura K. and Umino B. (1996). Methods for Automatic Term Recognition: A Review. *Terminology* 3(2), p. 259-289.
17. Lippincott, T., O Seaghdha, D., et al. (2010). *Exploring variation across biomedical subdomains*. In Proceedings of Coling. Beijing, China.
18. Mondary, T., Nazarenko, A., et al. (2012). The Quæro Evaluation Initiative on Term Extraction. In *Proceedings of LREC-8*, p. 663–669. Istambul, Turkey.
19. Nakagawa, H. and Mori, T. (2002). A simple but powerful automatic term extraction method. In *COMPUTERM 2002 – Proceedings of the 2nd International Workshop on Computational Terminology*, p. 29-35. Taipei, Taiwan.
20. Nakov, P. and Hearst, M. (2005). Using the web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of HLT-EMNLP*, p. 17–24.
21. Nédellec, C, Golik, W., et al. (2010). Building Large Lexicalized Ontologies from Text: a Use Case in Indexing Biotechnology Patents, *EKAW 2010.* Lisbon, Portugal, Oct. 11-15, 2010.
22. Nenadic, G., Okazaki, N., and Ananiadou, S. (2006). *Towards a terminological resource for biomedical text mining.* In *Proceedings of LREC-5*, Genoa, Italy, May.
23. Pazienza, M. T., Pennacchiotti, M. and Zanzotto, F. M. (2005). Terminology extraction: An analysis of linguistic and statistical approaches. In *S. Sirmakessis (eds.), Knowledge mining: Proceedings of the NEMIS 2004 final conference*, p. 255–279, Berlin Heidelberg. Springer.
24. Ratnaparkhi, A., Reynar, J. and Roukos, S. (1994). A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, p. 250 – 252.
25. Sant, P.M. (2004). Levenshtein distance. In *Dictionary of Algorithms and Data Structures* [online], Black P.E., (ed.), U.S. National Institute of Standards and Technology, 2004.
26. Santorini, B. (1990). *Part-of-speech tagging guidelines for the Penn Treebank Project*. Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.
27. Sclano, F. and Velardi, P. (2007). TermExtractor: a web application to learn the shared terminology of emergent web communities. In *Proceedings of I-ESA 2007*.
28. Schmid, H. (1994): Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK
29. Van Asch, V., & Daelemans, W. (2009). *Prepositional phrase attachment in shallow parsing*. In Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing (RANLP), p. 12-17. Borovets, Bulgaria: Association for Computational Linguistics
30. Vivaldi J. & Rodríguez H. (2007). Evaluation of terms and term extraction systems: A practical approach. *Terminology* 13:2, p. 225–248.
31. Wang, X., McCallum, A. and Wei, X. (2007). Topical N-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE international conference on data mining*, p. 697–702. Washington, DC. IEEE Computer Society.

32. Wermter, J. and Hahn, U. (2005). *Paradigmatic modifiability statistics for the extraction of complex multi-word terms*. In Proc. of HLT-EMNLP'05, 843–850.
33. Zargayouna, H. and Nazarenko, A. (2010). *Evaluation of Textual Knowledge Acquisition Tools: a Challenging Task.* In Proceedings of LREC 2010, p. 435–440, Valletta, Malta.
34. Z. Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. (2008). *A comparative evaluation of term recognition algorithms.* In Proceedings of LREC 2008.
35. Zhang, Z., Iria, J. and Ciravegna, F. (2010). Improving Domain-specific Entity Recognition with Automatic Term Recognition and Feature Extraction. In *Proceedings of LREC 2010*, p. 2606–2613. Valletta, Malta.

Knowledge-poor and Knowledge-rich Approaches for Multilingual Terminology Extraction

Béatrice Daille[1] and Helena Blancafort[2]

[1] University of Nantes, LINA, 2 Rue de la Houssinière,
BP 92208, 44322 Nantes, France,
beatrice.baille@univ-nantes.fr
[2] Syllabs, 53 bis rue Sedaine, 75011 Paris, France,
blancafort@syllabs.com

Abstract. In this paper, we present two terminology extraction tools in order to compare a knowledge-poor and a knowledge-rich approach. Both tools process single and multi-word terms and are designed to handle multilingualism. We run an evaluation on six languages and two different domains using crawled comparable corpora and hand-crafted reference term lists. We discuss the three main results achieved for terminology extraction. The first two evaluation scenarios concern the knowledge-rich framework. Firstly, we compare performances for each of the languages depending on the ranking that is applied: specificity score vs. the number of occurrences. Secondly, we examine the relevancy of the term variant identification to increase the precision ranking for any of the languages. The third evaluation scenario compares both tools and demonstrates that a probabilistic term extraction approach, developed with minimal effort, achieves satisfactory results when compared to a rule-based method.

1 Introduction

Identifying terms within a specific domain has been an active field of research since the early nineties [1,2]. Thanks to this research, several methods and tools have been developed for various applications such as information retrieval, information extraction, science and technology watch, and ontology acquisition. Twelve terminology extraction tools were described and compared in [3]. The various methods differ in how they process the corpus used as input, using anything from tokenization to syntactic analysis. Moreover, the tools differ in how they handle the output, e.g. with a manual validation process through dedicated interfaces. However sophisticated the processing of the input or output may be, all methods imply two steps that make up the core of the extraction process, namely:

1. **Step 1**: Identifying and collecting candidate terms (CT), i.e. term-like units in the texts (mostly multi-word phrases).
2. **Step 2**: Ranking the extracted CT to keep the most representative of the specialised field and the most useful for the target application.

Most of the tools are designed for one language with the exception of Termostat[4][3] which processes the Romance languages (French, English, Spanish, Italian and Portuguese) and Acabit[5][4] which works with French, English and Japanese. Concerning step 1, the CT can be made up of either multi-word terms only, as is the case with Acabit, or of both single (SWT) and multi-word terms (MWT), as is the case with Termostat. Acabit is the only tool that takes MWT variations into account i.e. the relation between basic and extended terms, as well as several forms of pattern switching. An example of pattern switching in French is the transformation of noun phrases with a Noun Adjective structure to a Noun Preposition Noun structure, e.g. as with the synonym terms *excès pondéral* $\leftrightarrow$ *excès de poids*, both meaning 'overweight'.
The objective of step 2 is to measure the termhood of a CT, i.e. the degree in which a CT is related to a domain-specific concept [6]. Several methods have been proposed for this task: the C-value method [7], based on the frequency of occurrence and term length weights the termhood of MWT according to their nested occurrences. The more an item is part of longer terms, the more it is likely to be a term. This measure applies only to MWT. But most of the work carried out uses statistical measures to compute the termhood of the CT. They are based on frequency counts and frequency distributions in the domain-specific corpora from which the CT are extracted [5]. Another research line compares the frequency of a CT in a domain-specific corpus and a language-general corpus [8,4]. The potential of different statistical measures (including an n-gram model) was evaluated by [9] to distinguish terms from non-terms in a CT list. They concluded that the number of occurrences ($freq$) is a very good indicator of the quality of a CT as well as the domain-specificity score (d_s).

The two terminology extraction tools that are presented in this paper encompass the main capabilities of current state-of-the-art tools. To complete step one, they handle SWT and MWT. In addition to this, the knowledge-rich method also processes SWT and MWT variation. To achieve the goal of step 2, they rank the CT according to specificity using a general language comparison corpus. The domain specificity d_s of a CT as defined by [8] is the quotient of its relative frequencies in both the monolingual comparable corpus (the domain corpus) rf_d and a general language corpus rf_g.

$$d_s(ct) = \frac{rf_d(ct)}{rf_g(ct)} = \frac{\frac{freq(ct)}{\sum_w freq(w)}}{\frac{freq(ct)}{\sum_{w'} freq(w')}} \quad (1)$$

Furthermore, the algorithms for steps 1 and 2 are formulated in a language-independent fashion. For the knowledge-rich approach, the language is a parameter: basic term and term variant patterns are formulated in terms of POS tags adopting the Multext POS tag annotations[5]. This language-independency allows

[3] http://olst.ling.umontreal.ca/~drouinp/termostat_web/doc_termostat/doc_termostat.html
[4] http://www.bdaille.com/
[5] http://aune.lpl.univ-aix.fr/projects/multext/

us to integrate a new language either by training the probabilistic tool, or by providing a defined set of language resources. In the next sections, we describe in detail the knowledge-poor approach, followed by the knowledge-rich approach.

2 Knowledge-poor Approach for Term Extraction

The knowledge-poor approach is based on a probabilistic tool. In contrast to a knowledge-rich tool that needs a POS tagger and hand-written rules to identify term candidates, the probabilistic tool simply requires a large raw corpus and a second smaller corpus with manually annotated sentences (noun phrases). This small corpus can be annotated by a linguist in a single day. The knowledge-poor approach is interesting for languages for which a POS tagger is not available. This can be the case when developing tools in an industrial context where open-source resources cannot be used because of license restrictions. It is also useful for under-resourced languages, for which annotated corpora are rare. It is possible today to compile a corpus for an under-resourced language from the web and use it as training material.

2.1 Training a Pseudo POS Tagger

Part-of-speech induction is the task of clustering words into word classes (or pseudo-POS) in a completely unsupervised setting. No prior knowledge such as a morphosyntactic lexicon or annotated corpus is required. The only resource needed is a relatively large training corpus. As in [10] and based on [11], we use Clark's tool[6] [12]. This tool for POS induction uses a distributional clustering algorithm and includes morphological information. The clustering algorithm is based on a cluster bigram model [13]. It is the highest performing system in almost every language, and one of the fastest methods. Performance and speed are important factors in an industrial context. The pseudo-POS tagger was trained using 50 clusters, after having run experiments with 20 to 100 clusters.

2.2 Corpora to Train the Pseudo POS Tagger

The input to the pseudo POS tagger is a tokenized corpus. For English, French and Spanish corpora we used the newstrain-08 corpora, monolingual language model training datasets which were provided for the WMT'09 translation task. Their size is approximately 2.5 GB for 500 million tokens (English), 1 GB for 175 million tokens (French), and 250 MB for 50 million tokens (Spanish). The German pseudo-POS tagger was trained on the German Wortschatz (350 MB for 60 million tokens), which performed a little better than the German newstrain-08 corpus. For Latvian we used a web-based corpus provided by an industrial partner.

[6] Available here: http://www.cs.rhul.ac.uk/home/alexc/pos2.tar.gz

2.3 CRFs to Train a Term Candidate Extractor

The terminological extraction task is close to the definition of the noun phrase chunking task, which is itself a subtask of the more general shallow parsing task. Traditional approaches in shallow parsing rely on a pre-processing step with a POS tagger. As in [10], the tool adopts the strategy of [14], who achieved near state-of-the-art results on the English supervised shallow parsing task using Conditional Random Fields (CRFs) [15]. CRFs enable a large number of features to be added in a flexible way. We used the CRF++ implementation, distributed under the GNU Lesser General Public License and new BSD License. The CRF model is trained on a tokenized corpus where sentences are separated by empty lines. Each line contains a word of the sentence together with its noun-phrase chunk tag. The tag is either B, I or O. B indicates the beginning of the noun phrase. I stands for inside the noun phrase. O represents tokens that do not belong to any phrase. In addition, the pseudo POS tag was used as one of the training features.

2.4 Training the CRF-based Term Candidate Extractor

To train the probabilistic CRF-based term candidate extractor, we used manually annotated corpora in each language. Small corpora with 300 to 600 sentences in French, English, Spanish and German were first automatically annotated with a symbolic term extractor and then manually corrected by a linguist. For Latvian, the corpus was manually annotated from scratch, as no rule-based system was available at that time. Table 2 gives more detailed information about the size and type of corpora used (general language corpus vs. domain-specific language corpus, raw corpus vs. manually annotated corpus).The domain-specific corpora were compiled using the focused web crawler Babouk [16].

3 Knowledge-rich Term Extraction Framework

The knowledge-rich approach requires linguistic knowledge to identify the CT (step 1). The following resources are needed:

- tools for the linguistic processing of the specialised texts: tokenizers, POS taggers and lemmatisers;
- hand-crafted patterns for the identification of single and multi-word CT based on POS tags;
- hand-crafted rules for the grouping of term variants.

The two first resources are mandatory, the last one is optional but needed to handle term variation.

For the linguistic processing step, we decided to use the TreeTagger[7] [17], because it performs both POS annotation and lemmatisation for 15 languages. The choice of the TreeTagger was thus determined by the number of languages available.

[7] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

Table 1. MWT: syntagmatic compounds of the noun category

	Pattern	Example	English translation
English	N N	*rotor blade*	
	A N	*renewable energy*	
French	N A	*énergie renouvelable*	*renewable energy*
	N S:p N	*caisse de résonance*	*sounding-box*
German	A N	*fossiler Energieträger*	*fossil energy source*
	N S:p N	*Netzintegration von Windenergie*	*grid integration of wind energy*
Latvian	A N	*meteoroloģiskā stacija*	*meteorological observing station*
	N:g N	*gaisa blīvums*	*air density*
Russian	A N	*метеорологический станция*	*meteorological station*
	N N:g	*выработка энергия*	*production of energy*
Spanish	N A	*energía eólica*	*wind energy*
	N S:p N	*fuente de energía*	*energy source*

3.1 Patterns for Candidate Term Identification

The term candidates and term variants are identified by means of patterns using Multext word classes. SWT are nouns or adjectives. MWT are noun phrases of length 2 or of length 3. A MWT of length 2 is a noun phrase with a head noun and a dependent of level 1, either an argument or a modifier (noun, adjective, etc.). To illustrate the patterns, the two main patterns for each language are provided (see Table 1).

3.2 Patterns for Term Variation

The term variant grouping functionality is optional and takes place once the CT has been annotated as a SWT or MWT. Several methods are implemented depending on the linguistic operation involved. There are 3 sub-functionalities: the detection of spelling term variants based on string distances, the detection of morphological variants based on monolingual lists of affixes and the detection of syntactic variants based on pattern rules on feature structures.

The spelling variants such as *air flow* ↔ *airflow* are detected by means of the edit distance. Morphological variants are handled by the Treetagger lemmatiser. Syntactic term grouping based on pattern rules consists in checking binary relation satisfactions between a pair of terms. For example, a binary relation is made between the MWT *énergie éolien* 'wind energy' and *énergie renouvelable éolien* 'renewable wind energy' according to the specifications of the French variant grouping pattern. The rule refers to a modification variant and expresses that a term whose components are a noun and an adjective should be related to any terms whose components are a noun and two adjectives, if and only if, they have the two same nouns and adjectives on the borders. Such a grouping pattern is written in a language-dependent grouping pattern specification file as follows:
Original term: N_0 A_1/ Variant: N_0 A_2 A_1
The term and the variant elements that are shared are numbered with the same

Table 2. Size of the domain-specific corpora in the domains of wind energy and mobile communication.

Language	De	En	Es	Fr	Lv	Ru
Wind energy: nb tokens	358,602	313,954	454,095	314,551	220,823	323,946
Mobile technologies: nb tokens	474,316	303,972	474,534	437,505	306,878	318,225

values. For example, in the above rule, the N_0 lemma of the term and of the variant are identical (same with A_1). The grouping patterns are thus not oriented: they are symmetric. The base term is defined as the most frequent of both items. Syntactic term grouping patterns cover the following syntactic phenomena: modification, coordination, compounding, decompounding. There is an average of 14 MWT patterns and 10 MWT grouping rules per language.

4 Resources

To assess the knowledge-poor and the knowledge-rich approaches, we use manually-checked comparable corpora, as well as hand-crafted reference term lists (RTL). The comparable corpora have been collected with a focused web crawler [16]. The corpus size varies from 300,000 to 400,000 tokens, depending on the domain and language. The RTL of around 130 terms in a specialised domain have been compiled to serve as a GOLD STANDARD for the evaluation of the tools. It should be noted that the wind energy domain corpora used to make our experiments are subsets of the corpora used to build the RTL with the exception of Latvian. For the mobile domain, the corpora are the same as those used for the compilation of the RTL with the exception of French, that differs slightly in terms of size.

4.1 Domain-specific Corpora

Because the tools for terminology extraction are particularly useful for new domains with poor terminological resources, the corpora used are related to two emerging domains: wind energy, a subdomain of renewable energy, and mobile technologies, a subdomain of computer science.

4.2 Reference Term Lists

The terms and variants are listed in the lemma form provided by the TreeTagger. The Reference Term Lists (RTL) were created manually to serve as a gold standard for the evaluation of the term extractors. They include both single (SWT) and multi-word terms (MWT) with their corresponding base terms and variants. One of the constraints is occurence in the corpus: a minimum term frequency of occurence was fixed, 5 for MWT and 10 for SWT. To decide on the termhood of

Table 3. Size of the RTL and corresponding corpora

Language	De	En	Es	Fr	Lv	Ru
Wind energy corpus: nb tokens	1,700,000	750,855	453,953	710,702	220,823	2,328,609
RTL size - Terms	132	128	136	126	129	107
RTL size - Variants	25	59	65	75	76	11
Mobile technologies: nb tokens	474,316	308,263	473,273	302,634	306,878	372,459
RTL size - Terms	159	140	137	130	139	103
RTL size - Variants	2	17	55	19	57	13

a term with respect to the domain, several linguistic criteria were applied [18]. Moreover, large terminology banks or specialised dictionaries (e.g. TERMIUM[8], Grand Dictionnaire Terminologique, IATE[9] and EuroTermBank[10] were used to check the terms.

Table 3 gives the RTL size expressed by the number of terms for each language and each domain. It recalls the size of the monolingual corpora used to build the RTL. The number of reference terms does not include the number of variants which are listed on a separate line. Compilation of these RTL is described in detail in [19]. For all the languages, we get an average of 130 RT. The number of variants depends on the languages. There are a higher number of variants encoded in the French, Spanish, English and Latvian lists, and nearly none for Russian for the wind energy domain. The wind energy domain has more variants than the mobile technologies domain. Spanish and Latvian display a large number of variants for both domains.

4.3 Reference Corpus of General Language

To calculate the domain-specificity (see equation 1), a general language corpus is needed. The general language corpora are a compilation of newspaper and Europarl data with 10 to 15 million words depending on the language. As an example, the German reference is based on the German newspaper TAZ and contains 20 millions tokens.

5 Results and Discussion

In this section, we evaluate the tools using the F-measure [20]. Precision is the percentage of the number of reference terms (RT) over the total number of canditate terms (CT) acquired from the corpus.

$$Precision = \frac{count_{RT=CT}}{count_{CT}} \quad (2)$$

[8] `http://termiumplus.gc.ca`
[9] `iate.europa.eu/`
[10] `http://www.eurotermbank.eu`, etc.

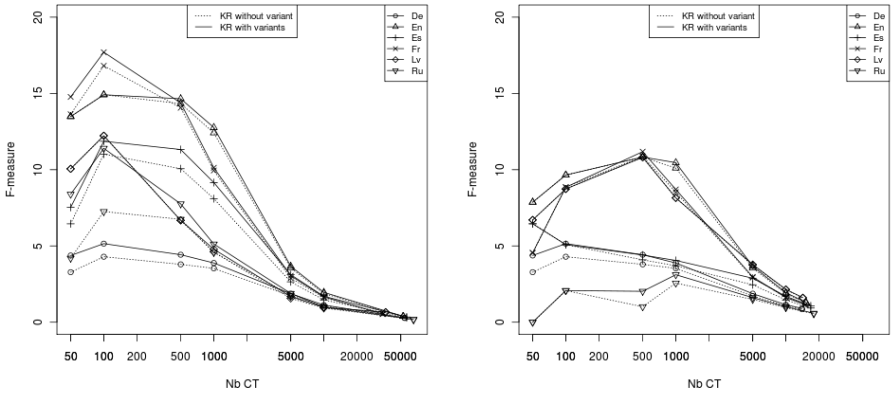


Fig. 1. F-measure ranking according to specificity (left) and the number of occurrences (right) for the wind energy domain

Recall is the percentage of the reference terms over the total number of terms from the reference term list contained in the corpus.

$$Recall = \frac{count_{RT=CT}}{count_{RT}} \tag{3}$$

The F-measure is defined as the harmonic mean of precision and recall. The F-measure has a value that is bound between 0 and 1, but we use here a percentage value.

$$F - measure = \frac{100 \cdot 2 \cdot precision \cdot recall}{precision + recall} \tag{4}$$

The first two evaluations focus on the knowledge-rich framework depending on two parameters (ranking and term variation), while the third evaluation compares both tools. The three evaluation scenarios are the following:

1. ranking of CT: the number of occurrences vs. the specificity (equation 1) (knowledge-rich framework);
2. term variants: the handling or not of term variation (knowledge-rich framework);
3. method: the knowledge-rich vs. the knowledge-poor approach

The CT ranking is given by the specificity value (d_s) or the number of occurrences ($freq$) in decreasing order. Table 4 illustrates the ranking of the CT alone and Table 5 with term variant recognition.

We consider a CT as correct if it matches a RT included in the RTL. The matches are made between lemmas. For the term extraction tools without variant recognition, we compare the CT and the RT. If they match, we return the CT rank. For the term extraction tools with variant recognition, we compare the CT or one of its variants and the RT, if either the term or one of its variants matches, we return the CT rank.

Table 4. CT ranking without variant recognition

Rank	Term or variant
1	wind project
2	wind energy project
3	aerodynamic
4	wind energy
5	wind turbine energy
6	onshore wind energy
7	energy from wind
8	small-scale wind energy

Table 5. CT ranking with variant recognition

Rank	Term or variant	Term or variant
1	T	wind project
1	V	wind energy project
2	T	aerodynamic
3	T	wind energy
3	V	wind turbine energy
3	V	onshore wind energy
3	V	energy from wind
3	V	small-scale wind energy

5.1 Ranking of Candidate Terms: Specificity vs. Occurrences

For the knowledge-rich approach, we compute the F-measure (see equation 4) and then we compare the ranking provided by the specificity and that provided by the number of occurrences for all the languages. We only use the terms of the RTL.

For all languages with the exception of German, and whether term variation is applied or not, the ranking of the specificity outperforms that of the number of occurrences. In a good ranking reflecting the termhood [6], the terms should appear on the top of the list: we clearly see a difference of shape between the ranking of the specificity and the occurrence. The higher results are obtained until the top 100 to top 500 candidates, with a clear decrease afterwards, although the ranking by occurrence does not show a stark contrast. This is striking for French, where the F-measure has nearly doubled from 9 to 17 points, as well as for Russian, where the F-measure rises from 2 to 12 for the top 100 CT. The only language for which there is no difference is German: this could only be explained by the empty intersection between the terms of the wind energy domain and the terms of the general language domain, which is the result of the compounding process. Figure 1 gives the F-measure on the wind energy corpus according to the specificity ranking for all the languages.

For the following two evaluation scenarios, we will only consider the ranking based on specificity.

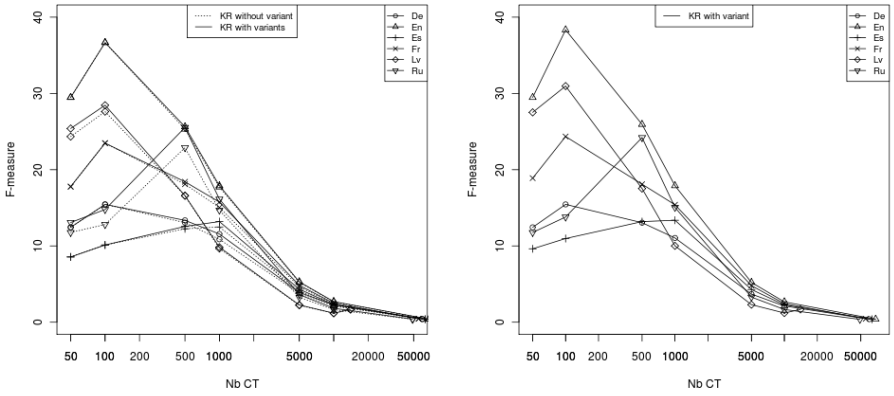


Fig. 2. F-measure ranking the CT (left) and the CT with variants (right) according to specificity for the mobile communication domain

5.2 Impact of Term Variation: with and without Variants

Here we examine the impact of term variant recognition on terminology extraction when using a knowledge-rich approach. When dealing only with the recognition of terms (see Fig. 1 left and 2 left), the detection of variants increases the F-measure for the highest ranks: this is the case for almost all languages with the exception of English and Latvian. We compare the recognition of terms alone using the RTL with terms only, and the recognition of terms and variants using the RTL with terms and variants. Fig. 2 shows the results obtained for both scenarios for the mobile domain. When we associate a set of synonymic variants to a term, the F-measure increases for almost all languages, with the exception of Russian (See Fig. 2 right). Looking through the Russian results, it appears that the recognition of terms increased but not the recognition of variants. This means that the knowledge-rich approach is able to correctly identify variants and that some variants appear in the list before the term to which they are related to. This result has however to be approached with care as RTL do not contain the same number and kind of variants.

5.3 Knowledge-poor vs. Knowledge-rich approach

In this subsection we compare the results provided by the knowledge-poor and the knowledge-rich approaches by examining the F-measure results obtained for the domains of wind energy and mobile communication. CT are ranked by the specificity score. Figures 3 show the results for English, Spanish and German. In English, the knowledge-poor method obtains similar results for both domains. The knowledge-poor approach performs better than the knowledge-rich approach for Spanish for the first 100 CT in the mobile domain but not for the wind energy

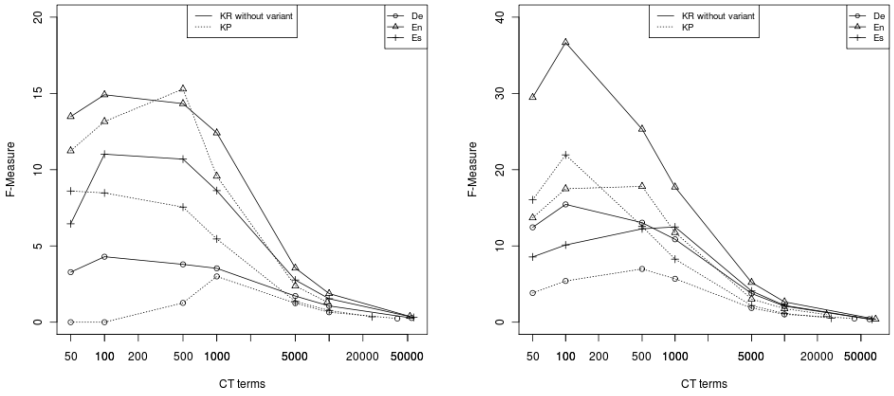


Fig. 3. F-measure ranked the CT according to specificity for the wind energy (left) and the mobile domain (right) either with knowledge-rich or knowledge-poor approaches

domain. For the other languages, French, German and Latvian, the knowledge-rich approach outperforms the probabilistic approach. However, the results for German are as low in both cases. This demonstrates the limits of the multilingual framework that applies the same symtagmatic approach for all languages. In addition to using syntactic patterns, morphological analysis is required for a language with productive compounding, e.g. German. In German, morphological compounds are much more frequent than MWT: 52% of nouns were reported to be compounds by [21] in the renewable energy domain. This means that a multilingual framework dedicated to terminology extraction should implement both morphological and syntactical processing. Concerning the knowledge-poor approach, the results are generally below the knowledge-rich approach. There are two reasons for this. First, the CT are not lemmatised, which is a severe obstacle for most of the languages, with the exception of English. As a matter of fact, the knowledge-poor approach delivers good results for English. Secondly, the knowledge-rich terminology extraction focuses on MWT of length 2 and 3, while the knowledge-poor approach, extracts MWT of unconstrained length. As the RTL do not include MWT with more than 3 tokens, longer MWT do not match with the terms in the RTL, e.g. the CT *small scale domestic wind turbine system* (Rank 428 in the CT list).

5.4 Related work

To our knowledge, no previous research has been done to use a probabilistic method for term extraction based on POS induction. An experiment on a similar task, namely on shallow parsing based on the English CoNLL 2000 corpus, is described in [10]. These experiments validate the knowledge-poor approach. They obtain interesting performances compared to a parser based on the Brill

[22] tagger. For the POS induction step, they use a corpus of 1 million words, and for the shallow parsing training, a few hundred annotated sentences. With the POS induction approach, they obtain an F-measure of 93.98 on noun phrase extraction against an F-Measure of 94.29 achieved with the Brill tagger. These experiments have been carried out in English. Work on CRF for shallow parsing is mainly carried out on English, probably because of the need to have a large amount of annotated data. [23] report work on chunking corpora using the Arabic Treebank and [24] report work on the UPENN Chinese Treebank-4. In our paper we include work on English, German and Spanish.

6 Conclusion

In this paper we have presented two terminology extraction tools that are designed to process a wide range of languages: a knowledge-poor and a knowledge-rich line. Both deal with SWT and MWT, and rank the CT according to domain specificity. The knowledge-poor approach is based on a probabilistic tool that performs pseudo POS tagging and thus could be an alternative for languages for which a POS tagger is not available. The knowledge-rich approach implements the main properties of state-of-the-art tools, and in addition handles term variation. Moreover, it is designed in a language-independent fashion: a language is a parameter where only term patterns and, optionally, term variation rules are required. We evaluated both approaches for two emerging domains and for six languages using hand-crafted reference term lists and manually-checked crawled comparable corpora. The results confirm that the specificity ranking outperforms the frequency of occurrence ranking and that the handling of term variants improves the ranking for the first candidate terms. Finally, the knowledge-poor approach provides satisfactory results with a minimal effort. In the future, it would be interesting to consider a scenario where a POS tagger is available and implement a method that uses a POS tagger but no hand-crafted rules, and then compare the results to the knowledge-rich and knowledge-poor tools that we have presented here.

References

1. Bourigault, D., Jacquemin, C., L'Homme, M.C., eds.: Recent Advances in Computational Terminology. John Benjamins (2001)
2. Kageura, K., Daille, B., Nakagawa, H., Chien, L.F.: Recent trends in computional terminology. Terminology **10**(2) (2004) 1–21
3. Cabré, M.T., Bagot, R.E., Platresi, J.V.: Automatic term detection: A review of current systems. In Bourigault, D., Jacquemin, C., L'Homme, M.C., eds.: Recent Advances in Computational Terminology. Volume 2 of Natural Language Processing. John Benjamins (2001) 53–88
4. Drouin, P.: Term extraction using non-technical corpora as a point of leverage. Terminology **9**(1) (2003) 99–117

5. Daille, B., Gaussier, E., Langé, J.M.: Towards automatic extraction of monolingual and bilingual terminology. In: Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-94), Kyoto, Japon (1994) 515–521
6. Kageura, K., Umino, B.: Methods for automatic term recognition: A review. Terminology **3**(2) (1996) 267–278
7. Frantzi, K.T., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the c-value/nc-value method. Int. J. on Digital Libraries **3**(2) (2000) 115–130
8. Ahmad, K., Davies, A., Fulford, H., Rogers, M.: What is a term? the semi-automatic extraction of terms from text. Translation Studies: An Interdiscipline (1994) 267–278 John Benjamins.
9. Drouin, P., Langlais, P.: Évaluation du potentiel terminologique de candidats termes. In: 8th Journées Internationales d'Analyse Statistique des Données Textuelles (JADT 2006), Besançon, France (2006) 379–388
10. Guégan, M., de Loupy, C.: Knowledge-poor approach to shallow parsing: Contribution of unsupervised part-of-speech induction. In Angelova, G., Bontcheva, K., Mitkov, R., Nicolov, N., eds.: RANLP, RANLP 2011 Organising Committee (2011) 33–40
11. Christodoulopoulos, C., Goldwater, S., Steedman, M.: Two decades of unsupervised pos induction: How far have we come? In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. EMNLP '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 575–584
12. Clark, A.: Combining distributional and morphological information for part of speech induction. In: EACL, The Association for Computer Linguistics (2003) 59–66
13. Ney, H., Essen, U., Kneser, R.: On structuring probabilistic dependencies in stochastic language modelling. Computer Speech and Language **8** (1994) 1–38
14. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. Technical Report CIS TR MS-CIS-02-35, University of Pennsylvania (2003)
15. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 282–289
16. de Groc, C.: Babouk: Focused Web Crawling for Corpus Compilation and Automatic Terminology Extraction. In: The IEEEWICACM International Conferences on Web Intelligence, Lyon, France (2011) 497–498
17. Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Actes, International Conference on New Methods in Language Processing. (1994)
18. L'Homme, M.C.: La terminologie : principes et techniques. Les Presses de l'Université de Montréal (2004)
19. Loginova, E., Gojun, A., Blancafort, H., Guégan, M., Gornostay, T., Heid, H.: Reference lists for the evaluation of term extraction tools. In: Proceedings of the Terminology and Knowledge Engineering Conference (TKE'2012). (2012)
20. Chinchor, N.: Muc4 evaluation metrics. In: In Proceedings of the 4th conference on Message understanding. MUC4 '92, Stroudsburg, PA, USA, Association for Computational Linguistics (1992) 22–29
21. Weller, M., Gojun, A., Heid, U., Daille, B., Harastani, R.: Simple methods for dealing with term variation and term alignment. In: In Proceedings of the 9th International Conference on Terminology and Artificial Intelligence, TIA 2011. (2011)

22. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics **4**(21) (1995) 543–565
23. Diab, M., Hacioglu, K., Jurafsky, D.: Automatic tagging of arabic text: from raw text to base phrase chunks. In: In 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04. (2004) 149–152
24. Chen, W., Zhang, Y., Isahara, H.: An empirical study of chinese chunking. In Calzolari, N., Cardie, C., Isabelle, P., eds.: ACL, The Association for Computer Linguistics (2006)

Committee-based Selection of Weakly Labeled Instances for Learning Relation Extraction

Tamara Bobić[1,2] and Roman Klinger[1,3]

[1] Fraunhofer SCAI, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
[2] B-IT, University of Bonn, Dahlmannstraße 2, 53113 Bonn, Germany
[3] Semantic Computing, CITEC, Bielefeld University, 33615 Bielefeld, Germany**
tamara.bobic@scai.fraunhofer.de
rklinger@cit-ec.uni-bielefeld.de

Abstract. Manual annotation is a tedious and time consuming process, usually needed for generating training corpora to be used in a machine learning scenario. The distant supervision paradigm aims at automatically generating such corpora from structured data. The active learning paradigm aims at reducing the effort needed for manual annotation. We explore active and distant learning approaches jointly to limit the amount of automatically generated data needed for the use case of relation extraction by increasing the quality of the annotations.
The main idea of using distantly labeled corpora is that they can simplify and speed-up the generation of models, *e. g.* for extracting relationships between entities of interest, while the selection of instances is typically performed randomly. We propose the use of query-by-committee to select instances instead. This approach is similar to the active learning paradigm, with a difference that unlabeled instances are weakly annotated, rather than by human experts. Different strategies using low or high confidence are compared to random selection. Experiments on publicly available data sets for detection of protein-protein interactions show a statistically significant improvement in F_1 measure when adding instances with a high agreement of the committee.

1 Introduction

Developing manually annotated training corpora for information extraction tasks like named entity recognition or relation extraction is tedious, time-consuming and therefore expensive work. One approach to overcome these issues is to build weakly supervised information extraction models, *e. g.* by using distantly labeled text, as proposed by [1]. This paradigm has shown to achieve reasonable, competitive results [2–4].

Unfortunately, the assumption that co-occurring entities in a sentence are related if they are mentioned in a source of distant supervision (for instance a database) does not hold in general. Therefore, such automatically annotated data sets are typically noisy. Methods addressing this issue include filtering approaches by formulating heuristics [5, 6] or classifying if the instance is actually representing a positive example [7]. In addition, though there is a huge amount of data available, the instances used for training may be uninformative and redundant.

** Present address of Roman Klinger. This work was performed at Fraunhofer SCAI.

In this paper, we explore and discuss the idea of making use of the active learning paradigm [8–10] to select meaningful distantly labeled instances from a large pool. Active learning is a strategy for reducing the overall annotation effort without diminishing the system's performance. It is a semi-automated approach where only data points that are considered to be most informative are presented to the "oracle" (usually a human expert) for manual annotation. We focus on the use case of classifying pairs of named entities as interacting or non-interacting. Objectives are to avoid using non-informative or misleading instances and to reduce the amount of data needed to train a model which leads to less complex models, as a lower number of features is generated.

Other approaches to circumvent the need for manual work include unsupervised machine learning approaches relying on discovering structure in unlabeled data. Although automatic generation of rule sets [11, 12], dictionaries [13, 14], or clusters [15] is effective, unsupervised approaches are often suffering from a limited performance in comparison to supervised approaches. Semi-supervised learning aims at obtaining good performance at a low cost by combining (potentially large) amounts of unlabeled data with human supervision. In the work by [16–18], a relatively small labeled seed set is used for learning initial patterns, while additional prediction rules are generated through further iterations. Such approach has the advantage of considerably reducing the amount of work for human annotators, however, due to its dependency on the initial seed set, the generalizability may be limited. A combined approach including both semi-supervised and active learning by [19] tends to increase the accuracy of label predictions, while keeping the human interference at minimum. In contrast, active learning aims at limiting the amount of work for a manual annotator. The fundamental idea is to make use of an estimator for selecting the instances to be shown to the annotator. That can be based on minimization of expected variance [20], uncertainty sampling [21], or query-by-committee [22], amongst others.

In the following, we shortly introduce interaction classification in Section 2.1 and explain how informative instances could be distinguished from redundant ones in Section 2.2. The results in Section 3 are based on evaluations of the proposed method on a publicly available data set for protein-protein and drug-drug interaction detection. We end with a discussion and summary.

2 Methods

2.1 Interaction Classification

As common, we formulate the task of relation extraction as feature-based classification of co-occurring entities in a sentence. Those are assigned to be either related or not, without identifying the type of relation. A sentence with n entities contains at most $\binom{n}{2}$ interacting pairs. We are using a linear support vector machine classifier [23] with lexical features, *i. e.*, bag-of-words and n-grams, with $n \in \{1, 2, 3, 4\}$. They encompass the local (window size 3) and global (window size 13) context left and right of the entity pair, along with the area between the entities [24]. Additionally, dictionary-based domain specific trigger words are taken into account. For details of the configuration, we refer to [5].

2.2 Committee-based Selection of Instances

We assume a small manually annotated training set to be available, the generation of which would require only a moderate amount of work. This set should provide the information for distinction between helpful and unhelpful or even misleading pairs of entities, *i. e.*, relation instances.

In active learning, a human annotator is asked to provide classification for an instance. The number of instances presented to the annotator is to be minimized, while the annotation is assumed to be perfect. Therefore, instances of highest expected information content (given the existing data) are normally chosen [25]. Here, we replace the human annotator by a predictor of limited knowledge, *i. e.*, the distantly labeled data. Retrieving high quality annotation for highly informative instances is possible from a human annotator, however, labels coming from the database are not always correct. Therefore, for the distantly labeled data the relation between the annotation quality and the information gain, given the seed set which is used for selecting instances is of importance. The hypothesis is, that a higher quality of the data annotation is positively correlated to a lower information gain. Therefore, in the distant supervision setting, there may be a trade-off between quality and information gain.

We follow several strategies to rank the instances and select the preferred ones. All strategies are based on a query-by-committee approach [26]. The training set for each committee member $c \in \mathcal{C}$ is selected by sampling n times with replacement, leading to approximate use of 63% of the available instances for each committee member [27] (where n is the number of available instances).

The agreement of the committee $\mathcal{C}$ concerning an instance i is measured as

$$u_{\mathcal{C}}(i) = \frac{|\mathbf{1}^i_{\mathcal{C}} - \mathbf{0}^i_{\mathcal{C}}|}{|\mathcal{C}|} ,$$

where $\mathbf{0}^i_{\mathcal{C}}$ denotes the number of committee members predicting "no interaction" and $\mathbf{1}^i_{\mathcal{C}}$ accordingly for predicting "interaction" for instance i.

High agreement of the committee is interpreted as high confidence regarding the label of an instance [22]. Let x_i be a random value from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with variance σ^2 and mean $\mu = 0$.

1. Rank descending by $u_{\mathcal{C}}(i)$ (prefer instances of high confidence).
2. Rank ascending by $u_{\mathcal{C}}(i)$ (prefer instances of low confidence).
3. Rank descending by $u_{\mathcal{C}}(i) + x_i$.

The idea of the first strategy is to select instances which are most similar (and therefore have a high quality) to the manually annotated training data, but may not lead to useful information. The second strategy pertains to the common approach used in active learning, where instances that are dissimilar to known ones may bring high information gain. The motivation of the third strategy is to take instances into account which are similar to the manually annotated data, but allowing the chance of having additional "novel" aspects.

3 Results

The silver standard corpora[4] of [29] consisting of 200,000 protein entity pair and 200,000 drug entity pair mentions are used as a source of weakly labeled data to draw training instances from. The text source are abstracts from MEDLINE[5]. They are labeled making use of the databases IntAct [30] and KUPS [31]. An overview of these corpora is given in Table 1.

Table 1: Weakly labeled PPI and DDI corpora.

	PPI	DDI
Abstracts	49,958	76,859
Sentences	51,934	79,701
Tokens	1,608,899	2,520,545
Entities	150,886	203,315
Pairs	200,000	200,000
Pos. Pairs	37,600	8,705

Table 2: PPI and DDI corpora.

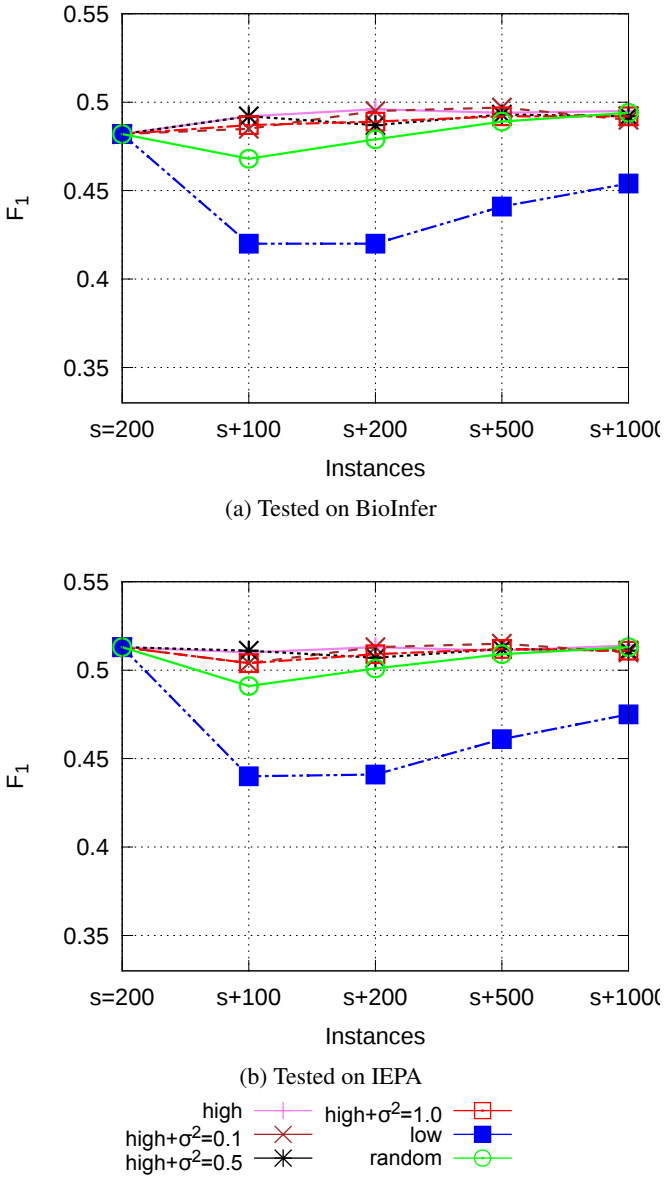
Corpus	Pos. pairs	Neg. pairs	Total
BioInfer	2,534	7,132	9,666
HPRD50	163	270	433
IEPA	335	482	817
LLL	164	166	330
DDI train	2,400	21,411	23,811
DDI test	755	6,275	7,030

The publicly available manually annotated corpora for protein-protein interaction HPRD50 [32], LLL [33], BioInfer [34], and IEPA [35] are used for training and testing. In case of drug-drug-interaction, the corpus published by [36] is used (being divided into train and test set). Table 2 shows an overview of the manually annotated PPI and DDI corpora.

The experimental setting is as follows. For each of the corpora, 200 pairs (instances) are randomly sampled from manually annotated data, corresponding to the seed set in an active learning setting. Based on these, classifiers are trained on sub-samples to predict if

[4] The term "silver standard" refers to an automatically annotated resource, contrary to a gold standard with (by definition) perfect annotation [28].

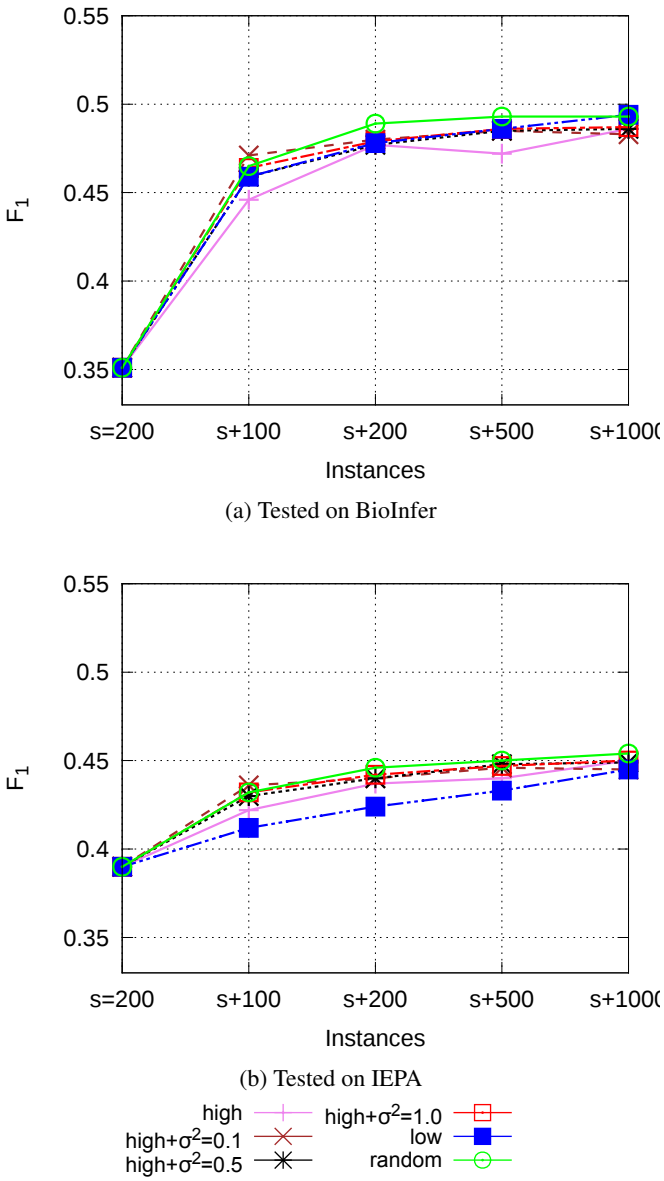
[5] `http://www.ncbi.nlm.nih.gov/pubmed/`



(a) Tested on BioInfer

(b) Tested on IEPA

Fig. 1: Confidence predictor based on LLL and testing on BioInfer and IEPA corpora. The model built on 200 manually annotated instances is compared against training with 100, 200, 500, and 1000 additional weakly labeled instances.



(a) Tested on BioInfer

(b) Tested on IEPA

Fig. 2: Confidence predictor based on HPRD50 and testing on BioInfer and IEPA corpora. The model built on 200 manually annotated instances is compared against training with 100, 200, 500, and 1000 additional weakly labeled instances.

an entity pair is in relation or not. This committee is used to get a score for agreement or disagreement of predicting data from the weakly labeled set. Depending on the selection strategy, 100, 200, 500, and 1000 instances are selected. A classifier is trained on the seed set of 200 instances, as well as on this set unified with the weakly labeled instances. Note that these are not multiple iterations, but separate experiments of active learning. Each of the experiments is repeated 10 times and the average value reported to be able to measure stability as well.

Figures 1 and 2 show the results for training on LLL and HPRD50, while testing on BioInfer and IEPA. The results between same models tested on different corpora are similar (compare 1a with 1b and 2a with 2b). In the case of training on LLL, worst strategy is selecting instances with the lowest confidence, followed by random. Best results are seen for the selection by high confidence, while adding Gaussian noise does not lead to big differences; for adding 100 weakly labeled instances, using $\sigma^2 = 0.5$ works best. All methods based on high confidence are outperforming the random baseline significantly in this step ($\alpha < 0.05$). Comparison of training with LLL and HPRD50 reveals notable differences when adding a low number of instances: for LLL, random and low confidence selection leads to a decrease. For HPRD50, all selection methods have a positive impact. Training on HPRD50 does not provide a clear difference between the selection strategies; low leads to worst results, random and high with some noise to the best. These differences are not significant.

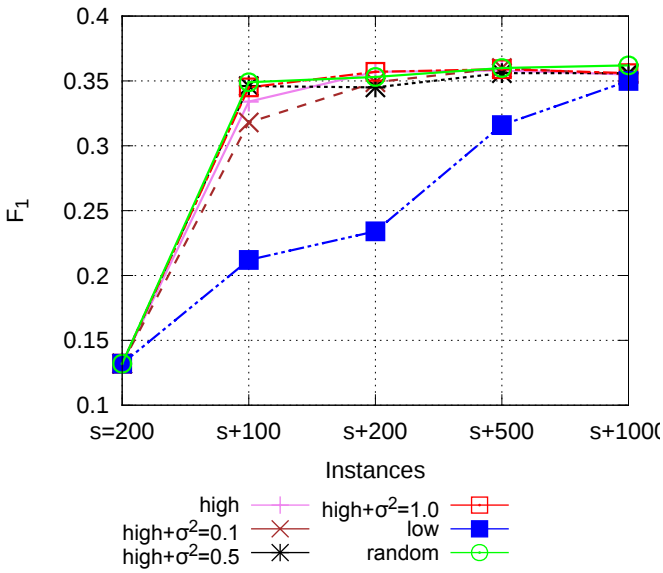


Fig. 3: Results for DDI, starting with 200 manually annotated instances and comparing against training with 100, 200, 500, and 1000 additional weakly labeled instances.

Obviously, adding a higher number of instances leads to lower impact of the selection strategy. Evaluating different strategies on DDI leads to results similar to training on HPRD50, as shown in Figure 3.

The results, especially for the seed set sampled from LLL (*cf.* Figure 1a), shows the best results using instances similar to the seed set (by means of the committee trained on the seed set having a high agreement). To prove the hypothesis that high quality of annotation is leading to a lower information gain and vice versa, the Pearson correlation coefficient of the committee prediction (based on a seed set of 200 LLL instances) and the labels from the database (the distantly labeled PPI corpus) are reported in Figure 4. For each confidence interval, 500 instances are sub-sampled respectively (1 refers to agreement among the committee; 0 refers to no agreement). A high correlation of the database labels for instances selected to be similar with the seed set can be observed. There is nearly no correlation for instances selected with low confidence.

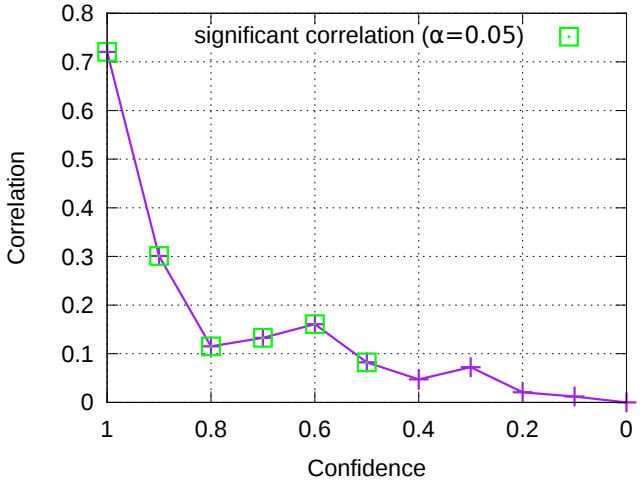


Fig. 4: Correlation of labels predicted by the committee and given from the database (distant supervision) plotted for different confidence threshold values. The correlation is calculated by Pearson's coefficient and the predictions are made using LLL as the training corpus.

4 Discussion and Summary

The results are motivating for a subset of corpora, shown on LLL here. The reason for the difference when training on HPRD50 is presumably the ratio between positive and negative examples; LLL is the most balanced set with a ratio of 1.01. HPRD50 has a ratio of 1.44. Due to the same reason, the initial results are that different (between 0.35

and 0.38 for HPRD50 in comparison to 0.48 to 0.51 for LLL) – the seed sub-sample only includes a low number of positive examples. Committee-based selection increases performance significantly on LLL.

It is notable for this corpus that the committee-based selection of weakly labeled instances leads to comparable results when using 100 additional instances chosen by high strategy and 500–1000 instances chosen randomly. Therefore it needs to be pointed out that, surprisingly, instances being similar to the seed set lead to best results. The reason is a strong correlation of database labels with the committee predictions in cases where the committee fully agrees.

Selecting instances labeled with the highest confidence by the committee appears to be the favourable decision in most cases to deal with the noisy data generated by the distant supervision approach. Such strategy is not common in the active learning paradigm, however, the prevailing in favor of "safe" instances confirms the hypothesis that a higher quality of the data annotation is correlated to a lower information gain.

It needs to be investigated further whether this methodology harms the generalizability of the model. An analysis of the positions of the support vectors from the seed set and from the weakly labeled set may allow insight in this concern.

Future work includes the evaluation of additional parameters. In comparison to active learning with a human annotator, additional knowledge about the weakly labeled data is available. Therefore, the ratio of positive and negative examples needs to be investigated further. Similarly, the characteristics of the seed set need to be analyzed in more detail. Furthermore, correlation of instances chosen to be in the seed set needs to be inspected, as well as the possible correlation between the seed set instances and those that are to be added.

Acknowledgments

T. Bobić was funded by the Bonn-Aachen International Center for Information Technology (B-IT) Research School. R. Klinger was partially funded by the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 248726.

References

1. Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. Proceedings of the International Conference on Intelligent Systems for Molecular Biology (1999)
2. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Conference of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. (2009)
3. Riedel, S., Yao, L., McCallum, A.: Modeling Relations and Their Mentions without Labeled Text. In: Proceedings of the European Conference on Machine Learning and Principles and Practice in Knowledge Discovery from Databases. (2010)

4. Thomas, P., Solt, I., Klinger, R., Leser, U.: Learning Protein Protein Interaction Extraction using Distant Supervision. In: Robust Unsupervised and Semi-Supervised Methods in Natural Language Processing, Recent Advances in Natural Language Processing. (2011)
5. Bobić, T., Klinger, R., Thomas, P., Hofmann-Apitius, M.: Improving distantly supervised extraction of drug-drug and protein-protein interactions. In: Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP, European Chapter of the Association for Computational Linguistics. (2012)
6. Buyko, E., Beisswanger, E., Hahn, U.: The extraction of pharmacogenetic and pharmacogenomic relations–a case study using PharmGKB. Pacific Symposium on Biocomputing (2012)
7. Yao, L., Riedel, S., McCallum, A.: Collective Cross-Document Relation Extraction Without Labeled Data. In: Empirical Methods in Natural Language Processing. (2010)
8. Olsson, F.: A literature survey of active machine learning in the context of natural language processing. Technical Report T2009:06, Swedish Institute of Computer Science (SICS), Kista, Sweden (2009)
9. Settles, B.: Active learning literature survey. Technical report, University of Wisconsin-Madison (2010) Computer Sciences Technical Report 1648.
10. Tomanek, K.: Resource-aware annotation through active learning. PhD thesis, TU Dortmund University (2010)
11. Brill, E.: Unsupervised learning of disambiguation rules for part of speech tagging. In: Proceedings of the Third Workshop on Very Large Corpora, Association for Computational Linguistics. (1995)
12. Hassan, H., Hassan, A., Emam, O.: Unsupervised information extraction approach using graph mutual reinforcement. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2006)
13. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. (1999)
14. Nadeau, D., Turney, P.D., Matwin, S.: Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In: Proceedings of the International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence, Berlin, Heidelberg, Springer-Verlag (2006)
15. Hasegawa, T., Sekine, S., Grishman, R.: Discovering relations among named entities from large corpora. In: Proceedings of the Association for Computational Linguistics. (2004)
16. Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the National Conference on Artificial intelligence and the Innovative Applications of Artificial Intelligence Conference. (1999)
17. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the ACM conference on Digital libraries. (2000) 85–94
18. Ravi, S., Baldridge, J., Knight, K.: Minimized models and grammar-informed initialization for supertagging with highly ambiguous lexicons. In: Proceedings of the Association for Computational Linguistics. (2010)
19. Wu, T., Pottenger, W.M.: A semi-supervised active learning algorithm for information extraction from textual data. Journal of the American Society for Information Science and Technology **56** (2005)
20. Cohn, D., Gharahamani, Z., Jordan, M.: Active learning with statistical models. Artifical Intelligence Research (1996)
21. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Empirical Methods on Natural Language Processing. (2008)
22. Freund, Y., Seung, S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning (1997)

23. Fan, E., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A Library for Large Linear Classification. Machine Learning Research **9** (2008) 1871–1874
24. Li, Y., Hu, X., Lin, H., Yang, Z.: Learning an enriched representation from unlabeled data for protein-protein interaction extraction. BMC Bioinformatics **11** (2010) S7
25. Nicholas, R., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: International Conference on Machine Learning. (2001)
26. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the workshop on computational learning theory. (1992)
27. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC (1993)
28. Rebholz-Schuhmann, D., Jimeno-Yepes, A.J., van Mulligen, E.M., Kang, N., Kors, J., Milward, D., Corbett, P., Buyko, E., Tomanek, K., Beisswanger, E., Hahn, U.: The CALBC Silver Standard Corpus for Biomedical Named Entities – A Study in Harmonizing the Contributions from Four Independent Named Entity Taggers. In: Proceedings of the Conference on Language Resources and Evaluation. (2010)
29. Thomas, P., Bobić, T., Hofmann-Apitius, M., Leser, U., Klinger, R.: Weakly labeled corpora as silver standard for drug-drug and protein-protein interaction. In: Workshop on Building and Evaluating Resources for Biomedical Text Mining, Language Resources and Evaluation Conference. (2012)
30. Kerrien, S., Aranda, B., Breuza, L., Bridge, A., Broackes-Carter, F., Chen, C., Duesbury, M., Dumousseau, M., Feuermann, M., Hinz, U., Jandrasits, C., Jimenez, R., Khadake, J., Mahadevan, U., Masson, P., Pedruzzi, I., Pfeiffenberger, E., Porras, P., Raghunath, A., Roechert, B., Orchard, S., Hermjakob, H.: The IntAct molecular interaction database in 2012. Nucleic Acids Research **40** (2012)
31. Chen, X., Jeong, J.C., Dermyer, P.: KUPS: constructing datasets of interacting and non-interacting protein pairs with associated attributions. Nucleic Acids Res **39** (2010)
32. Fundel, K., Kuffner, R., Zimmer, R.: RelEx–Relation extraction using dependency parse trees. Bioinformatics **23** (2007)
33. Nedellec, C.: Learning language in logic-genic interaction extraction challenge. In: Learning Language in Logic, International Conference on Machine Learning. (2005)
34. Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., Salakoski, T.: BioInfer: A Corpus for Information Extraction in the Biomedical Domain. BMC Bioinformatics (2007)
35. Ding, J., Berleant, D., Nettleton, D., Wurtele, E.: Mining MEDLINE: abstracts, sentences, or phrases? Pacific Symposium on Biocomputing (2002)
36. Segura-Bedmar, I., Martínez, P., Sanchez-Cisneros, D.: The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. In: Challenge Task on Drug-Drug Interaction Extraction 2011. (2011)

Suggestion Mining: Detecting Suggestions for Improvement in Users' Comments

Caroline Brun, Caroline Hagège

Xerox Research Centre Europe
6, chemin de Maupertuis, 38240, Meylan, France
Caroline.Brun@xrce.xerox.com, Caroline.Hagege@xrce.xerox.com

Abstract. In the context of the development of a feature-based opinion mining system for English, we observed that there is some very interesting information provided by customers but not yet covered by "standard" opinion mining techniques: opinion mining aims at detecting whether comments are positive or negative, but it appears that customers are very often suggesting improvements about what they are reviewing, which is quite different from expressing an opinion. This papers proposes to apply Natural Language Processing techniques in order to address this rather new task of extracting automatically such kind of suggestions for improvement from user's comments.

1 Introduction

With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise for information technologies to seek out and understand the opinions of others. There is a growing interest and activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in texts (see [1]). However, when looking at customer reviews in general, which are one of the core target of opinion mining applications, it appears that there is a wide part of very interesting information provided by customers but not yet covered by standard opinion mining techniques: opinion mining aims at detecting whether comments are positive or negative, but it appears that reviewers are very often suggesting improvements about what they are reviewing, which is quite different from expressing an opinion. At the same time, when someone in a review suggests some improvements for a given product, it means that this person is not fully satisfied with the current version of the product. This gives thus important information regarding the degree of satisfaction of a customer. Suggestions for improvement of a product are done using two discursive figures, either by wishing the presence of a missing feature or component, or by regretting the absence of such a feature or component. This goes beyond the scope of traditional sentiment analysis; however, it complements it and provides very valuable information that is worth collecting and analyzing in the context of business analytics. To illustrate this problem, the following

sentences are some instances of suggestions for improvement extracted from a corpus of user's comments about printers:

- "HP should have made the bin deep enough to hold an entire ream (500 sheets)."
- "Why HP didn't opt to throw in a 500-sheet tray with at least the standard 7 sizes (as per the 250-sheet tray) is beyond me."
- "This C4280 has filled the bill in those areas, however, for the price, HP should have considered throwing in more features and lowering the print costs."
- "I think they should have put a faster scanner on the machine, one at least as fast as the printer."
- "My only regret was that this unit could not fax!"
- "If you're considering any HP OfficeJet, I think most people will be happy with the R-40, which is only missing the automatic paper feed."

In this paper, we propose to use Natural Language Processing techniques in order to address the task of extracting automatically such kind of suggestions for improvement in comments. To perform this task, we have used a corpus of customer reviews about printers from different manufacturers.

2 Context of this Work

A fundamental task in opinion mining is classifying the polarity of a given text, sentence or feature/aspect level to find out whether it is positive, negative or neutral. Different methodologies using NLP and machine learning techniques are used for this purpose. Research has also shown that subjectivity or objectivity identification can also help achieving the purpose. However the most fine grained analysis model is the feature based sentiment mining method. Feature based opinion mining aims at determining the sentiments or opinions that are expressed on different features or aspects of entities (e.g. [2],[3], [4]). When a text is classified at document level or sentence level, it might not tell what the opinion holder likes or dislikes. If a document is globally positive about an object, it does not mean that the opinion holder holds positive opinions about all the aspects or features of this object. Similarly, if a document is globally negative, it does not imply that the opinion holder dislikes everything about the object described.
The context of this paper is the development of a feature-based opinion mining system, for English.
Formally, our system adopts the representation of a given opinion as proposed by [1], where an opinion is a five place predicate of the form $(o_j, f_{jk}, s_{ijkl}, h_i, t_l)$, where:

- o_j is the target object of the opinion (the main concept)
- f_{jk} is a feature associated to the object
- s_{ijkl} is the value (positive or negative) of the opinion expressed by the opinion holder about the feature

- h_i is the opinion holder
- t_l is the time when the opinion is expressed.

The opinion detection system we have built relies on a robust deep syntactic parser, (c.f. [5]), as a fundamental component, from which semantic relations of opinion are calculated. Once syntactic relations are extracted by a general dependency grammar, we use the robust parser by combining lexical information about word polarities, sub-categorization information and syntactic dependencies to extract these semantic relations. The polarity lexicon has been built using existing resources and also by applying classification techniques over large corpora, while the semantic extraction rules are handcrafted, see [6] and [7] for the complete description of these different components. There exist other systems, such as the one described in [3], that use syntactic dependencies to link the source and target of the opinions. Our system belongs to this family, since we believe that the syntactic processing of complex phenomena (negation, comparison and anaphora) is a necessary step in order to perform feature-based opinion mining. Another characteristic of our system is that it respects a two-level architecture; it relies on a generic level, applicable to all domains and corpora, and on a domain-dependent level, adapted for each sub-domain of application.
Additionally, our system includes a semantic mapping between polar vocabulary and the features it corresponds to. For instance, the opinion word "fast" is mapped to the feature "speed", the word "expensive" to the feature "price", the word "clunk" to "noise" and so on. This mapping enables us to further exploit the comments of the user by referring to specific product characteristics.

Some examples follow that show what the system would output considering certain input sentences extracted from customer reviews about printers:

1. *Input*: "I like this printer, but I think it is too expensive."
   Output: OPINION(printer, _, positive, I, _),
   OPINION(printer, price,negative, I, _).

   In this example, the system identifies that the user expresses a negative opinion on the feature "price" of the printer although the rest of its characteristics satisfies him.
2. *Input*: "The problem of this printer is the fuser."
   Output:
   OPINION(printer, fuser, negative, _, _).

   Here, the system identifies that the user expresses a negative opinion about the feature "fuser" of the printer.

In this paper, we focus on the integration of a new component to this preexisting system, which is a component aiming at detecting suggestions for improvement within customer reviews and that goes beyond the scope of traditional opinion detection.

3 Related Work

While there is a lot of literature regarding opinion mining and more precisely feature-based opinion-mining, we didn't find many works related to the automatic extraction of suggestions within comments, except the following paper by [8] about extracting wishes, which has some connections with the approach described here. The cited paper presents a work whose goal is related to our goal: it aims at detecting wishes in documents using NLP techniques combined with Machine Learning techniques. Their tool relies on very simple pattern matching (about 13 patterns like "I wish _", "if only _", "would like _ if", ...) combined with pattern-based sentence classification (using Support Vector Machines). They train their system using a corpus of New Year's wishes made by people around the world (see http://www.timessquarenyc.org/events/new-years-eve/wishing-wall/index.aspx). The goal they address is wider than the one proposed here on one hand, and more restricted on the other. As they extract all kind of wishes, the scope is wider than ours, since suggestions for improvement can be viewed as subset of wishes: for example, from our customer printer's reviews, their method would extract "I would like to share my experience", "what I would like to stress here in this review", (these expressions are wishes indeed) while our method would filter out these sentences since they are not suggestions for improvement (and not very useful in a business analytics context). Moreover, as they are using a corpus containing exclusively (very general) new year's wishes, their system is not covering at all the other way people are expressing suggestions, i.e. by regretting the absence of a feature or a component: our method would extract "My only regret was that this unit could not fax!", "The lack of any PC fax status software is incredible", which are implicit suggestions for improvement, while their proposal is not covering these cases.

4 Detecting Suggestions

As said in introduction, suggestions appearing in user comments are expressed using "wishes" or "regret". Therefore, our suggestion detection component relies on syntactico-semantic patterns that capture these types of expressions. The global architecture is showed on figure 1.

The system relies on four main components:

- a structured terminology of the target topic;
- a thesaurus of vocabulary related to expressions of suggestions;
- a fine grained linguistic parser;
- an extractor for suggestions for improvement (SfI).

These different components are detailed in the following sections.

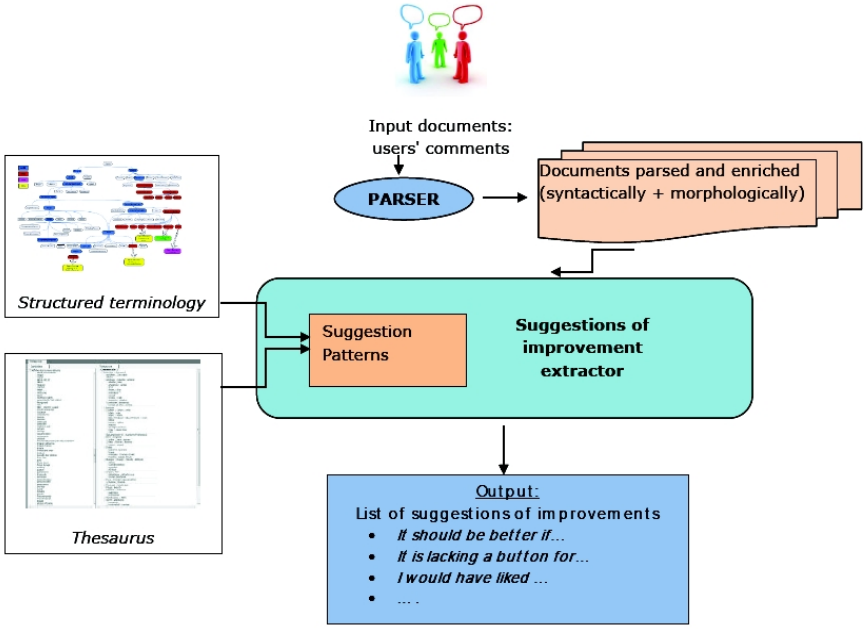


Fig. 1. Architecture of the system

4.1 Structured Terminology

The structured terminology consists in a description of the topic in focus. Taking for example a manufactured product like a printer, the terminology encodes concepts and associated terms such as the "part-of" concepts of the product (for instance *paper tray*), the manufacturer of the product (for instance *Xerox*), physical characteristics (for instance *weight*, *aspect*) and commercial/technical characteristics (for instance *speed*, *price*, *noise*). The terms found in the terminology are potential targets on which suggestions for improvement may apply. These terminologies can be part of already existing ones if available (e.g. in an existing opinion mining system) or can be acquired from processing texts of the domain as very simple ontological relations are needed. In our case, the structured terminology was already part of our opinion detection system, since this information is also needed to perform feature-based opinion mining. To extract this domain-dependent terminology, we have used a method partly similar to what is proposed in [9], who seek meronymy relationships (part-whole). In our case, the main concepts discussed in reviews about printers are denoted by the following vocabulary : *printer*, *copier*, *scanner*, *machine*, and *product*. We have used our parser to extract, from the corpus, the most frequent nouns modifying these main concepts, i.e. matching the MODIFIER syntactic relations and calculate frequencies of these relations.

4.2 Thesaurus of Expressions of Suggestion

The thesaurus of expression of suggestions is encodes expressions used in sentences denoting wishes, regrets and misses. Choosing as seeds the words *wish*, *regret*, *miss* and *lack* we explored existing thesauri and gather nouns and verbs that will potentially trigger expressions of suggestion. Existing thesauri can be found for instance in the following links : http://thesaurus.com/, http://www.macmillandictionary.com/, etc.. We finally consider two different semantic classes for verbs and predicative nouns. The class *absence* which contains verbs such as *lack*, or noun like *absence*, the class *belief* which contains verbs like *expect* or *believe*. We gather a total of 16 lexical entries of classes *absence* and *belief* and also consider isolated lexical entries such as *wish* considered as verb or noun, the verb *like* and the verb *prefer*.

4.3 Linguistic Processor

As said in section 2, the linguistic processor we used is a robust syntactic parser, that performs a fine grained linguistic analysis of input text. This analysis contains the morphological features of the words (in particular information about modality and aspect) but it also gives us dependency links between the linguistic units in the text. These dependencies are labeled with the main syntactic functions such as *SUBJECT*, *OBJECT*, *MODIFIERS* (see [5] for more details). Automatic detection of named entities is also handled by this system, and covers the extraction of standard named entity type such has PERSON, COMPANY, PRODUCT NAMES, PLACES, DATES ...
Input texts (i.e users reviews) are processed by the linguistic processor which will calculate the syntactic relations holding between the words and will give also information about verbal tenses, modality and verbal aspect. For instance, the following sentence *I would have liked a better resolution for my pictures*, the linguistic analysis provides the following information.

```
MOD_PRE(resolution,better)
SUBJECT(like,I)
VMODAL(like)
VASPECT_PERFECTIVE(like)
VTENSE_PAST(like)
OBJECT(like,resolution)
PRINTER_FONCTION(resolution)
```

The system has detected that the verb *like* in this sentence is used together with a modal form and is employed in a perfective aspect. It also has detected that the object complement of *like* is the word *resolution* which is a printer function. Finally, the system also detects that the subject of *like* is the pronoun *I*.

4.4 Extractor of Suggestions for Improvement

The core of the extractor of suggestions for improvement consists in a set of general syntactico-semantic patterns that will then be matched against exist-

ing analyzed reviews. These patterns express general formulas of suggestion in which arguments are elements of the domain terminology. These patterns also use suggestion expressions that have been extracted from the thesaurus. Finally, linguistic information such as dependency relations, verbal modality and aspect is also enclosed in the patterns. For instance, in the context of printers' reviews, the following syntactico-semantic pattern uses both linguistic and terminological information and expresses a suggestion for improvement.

$$\begin{bmatrix} \text{SUBJECT} & \left[\text{SEMANTIC-TYPE} = \text{MANUFACTURER}\right] \\ \\ \text{PREDICATE} & \begin{bmatrix} \text{TENSE} = & \text{PAST} \\ \text{ASPECT} = & \text{PERFECTIVE} \\ \text{M}\mathit{odal} = & \text{YES} \end{bmatrix} \end{bmatrix}$$

This pattern can be rephrased as "a manufacturer entity, which is subject of a modal verb used in the past tense and perfective aspect". This pattern, which corresponds semantically to a wish, extracts the pertinent sentence *HP should have made the bin deep enough*, but does not extract the non-pertinent sentence *HP should be ashamed*, as *ashamed* is not a verb in the perfective form.

The following pattern can be rephrased as "a product-function term, which is subject of a verb of the semantic class absence".

$$\begin{bmatrix} \text{SUBJECT} & \left[\text{SEMANTIC-TYPE} = \text{PRODUCT-FUNCTION}\right] \\ \\ \text{PREDICATE} & \left[\text{SEMANTIC-CLASS} = \text{ABSENCE}\right] \end{bmatrix}$$

This pattern does not correspond to a wish as in the previous example. It just states that something is missing in the described product, which implicitly means that the user would have liked to have this component. It extracts (among others) the pertinent sentence *But the ability to kill jobs is really lacking, often requiring hard reset when needed.*

Such patterns are applied after the analysis of the customer reviews by our robust parser. When a given pattern matches a sentence analysis result, the sentence is considered as relevant. The output of the system consists in a list of sentences expressing either customer wishes regarding the products or absence of components or functionalities. This result can then be manually or semi-automatically processed in order to detect what are the main claims of the customers without having to send them any questionnaire.

We defined 21 patterns for our SfI. It is worth mentioning that some of the patterns also make use of polarity lexicon defined for the opinion-mining application. For instance the following pattern states that if a negative adjective which is used with a comparative of inferiority is also an attribute of a product characteristics (terminology information), and if this product characteristics is the subject of a modal verb, then we extract the sentence. This pattern will

cover examples as *fax could be less noisy.*

$$\begin{bmatrix} \text{SUBJECT} & \left[\text{SEMANTIC-TYPE} = \text{PRODUCT-CHARACTERISTICS}\right] \\ \text{PREDICATE} & \begin{bmatrix} \text{MODAL} = & \text{YES} \\ \text{ATTRIBUTE} & \begin{bmatrix} \text{POLARITY} = & - \\ \text{COMP} = & \text{INF} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Finally, it is also important to note that suggestions for improvement are not necessarily correlated with positive/negative opinions. Suggestions can be expressed in texts where the general opinion is either positive or negative. Even at the sentence level, suggestions can appear in a positive sentence like *But the lack of this feature isn't a complaint, it's a request for the next model*, even if they appear more often in neutral or negative comments as in *Unfortunately many of the manuals are only available on-line as I would have liked a printed main manual.* At the same time, suggestions for improvement can be considered as slightly negative (if someone suggests an improvement for a product, it means that he/she is not completely satisfied with the product). As a result, integrating the suggestions for improvement module in a more general opinion-mining tool reflects more precisely the customer's opinion, and complete it, providing useful information to the manufacturer.

5 Evaluation

5.1 Corpus

In order to build our opinion detection system, we used a corpus of reviews available on the website "Epinion"[1]. This is a general site compiling millions of user reviews about products, movies, books, etc. These reviews are semi-structured and contain the following information:

- The product name;
- the overall score (from 0 to 5 stars): we consider that reviews ranked from 0 to 2 are globally negative and that reviews from 3 to 5 stars are globally positive;
- the review title;
- the creation date;
- the sections "Pros", "Cons" and "Bottom Line";
- the content of the review in free text, with the assessment: "Recommended": "yes" or "no".

As our first target application deals with consumer reviews about printers, we extracted a corpus of about 3,500 printer reviews from this site. These reviews in html format have been cleaned and converted into xml format, ready for processing.

[1] http://www.epinions.com/

5.2 Experimental Results

We selected an unseen part of the corpus previously described in order to perform the evaluation of the SfI detector prototype we developed. The main problem was that this corpus was not annotated in terms of suggestions from the user. We have therefore extracted randomly 60 reviews from our initial corpus and annotated them manually. After running the prototype against this sub-corpus we obtained the following measures (table 1):

Table 1. Evaluation results

Precision	Recall	F-measure
77%	70%	73%

These results are quite satisfying regarding the complexity of the task, as the approach we adopted frequently catches suggestions from users which can lead to product improvements.
However, regarding precision, some erroneous suggestions were extracted. Among them, we can find ironic suggestions like in *Perhaps they should have stuck with making sewing machines and typewriters* or in *I soon discovered, however, that for the money they spent on this jammin' little number, they should have spent a couple of nights on the beautiful Oregon Coast and pampered themselves at expensive restaurants instead.*. At the moment, we are unable to automatically detect irony, and, apart from this, these sentences match perfectly our formal criteria.

Furthermore, our system also detects erroneously complex expressions where a suggestion is present just to reinforce customer satisfaction, as for instance in *...in my opinion, the lack of high end features is a very fair trade-off for the outrageously low price tag.* Our system also detects suggestions expressed in beliefs like in *Because of previous HP experiences, I knew the printer would work great, but I figured the other features would lack in quality.*, which can not be considered as a suggestion for improvement.

Regarding recall, our system sometimes misses interesting suggestions for improvement mentioned by the users in the reviews. Here are some examples of this kind of errors.
We did not consider that interrogative forms like *Why not just merge into one easy-to-use program?* could sometimes correspond to suggestions for improvement. In this sentence, although the work *program* belongs to the terminology related to the printer and the adjective *easy-to-use* is detected as a positive adjective related to *program*, we have no suggestions template stating that interrogations around a positive aspect of a printer attribute can correspond to a suggestion. This kind of patterns can be added however to the system. Another source of errors is related to the capability of the parser to deal with colloquial language as for instance in *it woulda been nice if this model had an ethernet link*

built in. Finally, we noticed that our system was not able to deal with vagueness in suggestions: sometimes previous textual context makes clear what the user wants, but locally, at sentence level, the improvement needed remains very vague. This is for example the case of *Backlinghting on the LCT screen would help this thing so much*.

5.3 Potential Extension

As said before, we had not corpora annotated in term of suggestions from the users. However, many companies nowadays propose users to express what they want to be improved in products, either by filling some forms[2] or by interacting on some dedicated web forums[3].
This shows how useful automatic suggestion detection would be for these companies, but also potentially constitute a very valuable corpus for the development of such an application. One of the perspective of our work is to use this kind of user-generated corpus to simulate a manually annotated corpus in term of suggestions (hopefully with a minimal human intervention).
From such a corpus, we plan to develop an automatic classifier for sentences containing suggestions for improvement. The idea is to use the different patterns presented in this paper as features for the classifier. We have previously experimented Support Vector Machines based on linguistic features for positive or negative review classification, so we plan to apply the same kind of technique for suggestion classification.

6 Conclusion and Perspectives

We have developed an approach that automatically detects suggestions for improvement expressions from customer reviews. This approach relies on linguistic knowledge and is combined with an opinion mining system. It enriches the opinion mining module if we consider that when a user suggests improvements for a product it means that (s)he is not entirely satisfied with this product. Furthermore, analysis of suggestions from customers can be of great help for the product manufacturers. And additionally, such suggestions, when automatically extracted from customer reviews can also be very useful to improve recommendation systems (see [10]).
Our future perspective is to enrich the patterns for suggestions for improvement, in particular to be able to deal with suggestions expressed in the interrogative form. Furthermore, as this detection module is integrated in a feature-based opinion detection system, we also want to further refine the extraction rules, in order to precisely detect the targets of the suggestions. The plan is also to build an automatic classifier of sentences containing suggestions using these linguistic patterns as features.

[2] e.g. http://www.sonycreativesoftware.com/support/productsuggestion.asp
[3] e.g. http://community.norton.com/t5/Product-Suggestions/bd-p/Suggestions

References

1. Liu, B.: Sentiment analysis and subjectivity. Handbook of Natural Language Processing **2nd ed** (2010)
2. Bloom, K., Garg, N., Argamon, S.: Extracting appraisal expressions. In: HLT-NAACL 2007, Rochester, NY (2007) 308–315
3. Kim, S.M., Hovy, E.: Identifying and analyzing judgment opinions. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. HLT-NAACL '06, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 200–207
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD 04, ACM (2004) 168–177
5. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond shallowness: incremental deep parsing. Nat. Lang. Eng. **8** (2002) 121–144
6. Brun, C.: Detecting opinions using deep syntactic analysis. In: RANLP. (2011) 392–398
7. Brun, C.: Learning opinionated patterns for contextual opinion detection. In: Proceedings of the 24th Conference on Computational Linguistics. COLING 2012, Mumbai, India (2012)
8. Goldberg, A.B., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B., Zhu, X.: May all your wishes come true: a study of wishes and how to recognize them. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 263–271
9. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. 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) 339–346
10. Stavrianou, A., Brun, C.: Opinion and suggestion analysis for expert recommendations. In: Proceedings of the Workshop on Semantic Analysis in Social Media, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 61–69

Automatic Detection of Outdated Information in Wikipedia Infoboxes

Thong Tran[1] and Tru H. Cao[2]

[1] Da Lat University and John von Neumann Institute - VNUHCM
thongt@dlu.edu.vn

[2] Ho Chi Minh City University of Technology and
John von Neumann Institute - VNUHCM
tru@cse.hcmut.edu.vn

Abstract. An infobox of a Wikipedia article generally contains key facts in the article and is organized as attribute-value pairs. Infoboxes not only allow readers to rapidly gather the most important information about some aspects of the articles in which they appear, but also provide a source for many knowledge bases derived from Wikipedia. However, not all the values of infobox attributes are updated frequently and accurately. In this paper, we propose a method to automatically detect outdated attribute values in Wikipedia infoboxes by using facts extracted from the general Web. Our method uses the pattern-based fact extraction approach. The patterns for fact extraction are automatically learned using a number of available seeds in related Wikipedia infoboxes. We have tested and evaluated our system on a set of 100 well-established companies in the NASDAQ-100 index on their employee numbers, presented by the *num_employees* attribute value in their Wikipedia article infoboxes. The achieved accuracy is 77% and our test result also reveals that 82% of the companies do not have their latest numbers of employees in their Wikipedia article infoboxes.

Keywords: Information Extraction, Wikipedia Update, Pattern Learning.

1 Introduction

Currently, Wikipedia has grown into one of the central knowledge sources of mankind. Since its creation in 2001, Wikipedia has become one of the most popular websites in the world. As of August 2012, the English version of Wikipedia contains almost 4 million articles. The infobox of a Wikipedia article contains important facts about that article that are summarized in the tabulated form. Thanks to such structured information, readers can rapidly gather the most important information about some aspects of the article. Moreover, information in infoboxes can be automatically extracted and reorganized into ontologies, knowledge bases, or databases, such as DBPedia [1], YAGO [2], and Freebase [3].

All the Wikipedia's content is manually updated or maintained by contributors. This leads to the fact that its information is not updated regularly and completely,

while there is continuous change of related information on the Web. So, it is essential to have automatic methods to update Wikipedia's content to ensure that it contains the latest information. For instance, Fig. 1 is the infobox in the Wikipedia article of company *Activision Blizzard*. It shows that the number of employees of this company is *5,000* in *2011* and this value has not been updated until now[1]. However, the facts from three snippets of web pages[2,3] in Fig. 2 show that its current number of employees has changed to *7,300* since *December 31, 2011*. That means the current value of the number-of-employees attribute in the infobox of *Activision Blizzard* is outdated.

Fig. 1. The current infobox of company Activision Blizzard in Wikipedia

Fig. 2. The number of employees of company Activision Blizzard found on the Web

Wikipedia can be enriched using two main sources of information that are Wikipedia articles themselves or external sources such as the Web or some other knowledge bases. In [4], Lange et al. proposed a method for automatically populating infoboxes of Wikipedia articles by extracting unstructured information from the content of their articles. It used Conditional Random Field [5] for training to extract attribute values. Catriple [6] was a system that automatically extracted triples from Wikipedia articles and non-isa properties from Wikipedia categories. Those triples

[1] The time when this paper is completed.
[2] http://www.insideview.com/directory/activision-blizzard-inc
[3] http://www.alacrastore.com/storecontent/MarketLine_formerly_Datamonitor_Company_Profiles-Activision_Blizzard_Inc-2123-27054

could be used to add new attributes of an infobox. Meanwhile, Kylin [7] was a system that created new infoboxes or filled up existing infoboxes of Wikipedia articles by using articles with similar infoboxes to determine common attributes. The attribute values in those infoboxes were then extracted from their containing articles. Using an external source, Syed et al. [8] proposed a method that used information from Wikitolog [9], a hybrid knowledge base of structured and unstructured information, beside existing Wikipedia articles, for adding categories and inter-article links to new Wikipedia articles. Wu et al. [10] extended system Kylin [7] using out-of-Wikipedia information extracted by TextRunner [11]. However, those enrichment methods and systems only focused on adding new data or filling incomplete data to Wikipedia, but not detecting and updating its outdated information.

Besides, our proposed method involves information extraction on the Web, in particular relation extraction. As bootstrap-based systems, DIPRE [12] and Snowball [13] used a seed set of examples of the relation to be extracted for initialization, and then iteratively learned patterns and extracted instances of that relation. With the self-supervised approach, KnowItAll [11], TextRunner [14], and SRES [15] were systems that used only a few labeled examples to learn to extract relations from Web. Mintz et al. [16] investigated an alternative paradigm for automatic relation extraction without requiring labeled corpora, avoiding domain dependence. In contrast to those research works, relation extraction in our proposed method is driven by target outdated infoboxes in Wikipedia.

The remainder of this paper is structured as follows. Section 2 details our proposed method. Section 3 presents experiments and evaluation of the proposed method. Finally, Section 4 concludes the paper and suggests future work.

2 Proposed Method

2.1 Definitions

For clarity, we define the basic notions that are used to present the proposed method.

Definition 2.1: *Fact*
A fact f on a binary relation r, denoted by f_r, is defined to be of the form $<r(e_1, e_2), t>$ where e_1 and e_2 are entities of an instance of r, and t is the associated time of the fact.

We use $f_r.entity1$, $f_r.entity2$, and $f_r.time$ to denote the first entity, the second entity, and the time, respectively, of the fact f_r. For example, with the fact $f_{NumOfEmployees}$ = <*NumOfEmployees*(*Activision Blizzard, 5000*)*, 2011*> extracted from the infobox of company *Activision Blizzard* as shown in Fig. 1, $f_{NumOfEmployees}.entity1$ = *Activision Blizzard,* $f_{NumOfEmployees}.entity2$ = *5,*000, and $f_{NumOfEmployees}.time$ = *2011*. This fact says that the number of employees of company *Activision Blizzard* is *5,000* in *2011*.

Definition 2.2: *Outdated fact*
A fact f is defined to be outdated with respect to another fact g on the same relation r, if and only if:

1. $f_r.entity1 = g_r.entity1$, and
2. $f_r.entity2 \neq g_r.entity2$, and
3. $f_r.time < g_r.time$

Without loss of generality, in the above definition, we assume the outdated information is always at the second entity of a relation of discourse. Also, for an outdated fact in a Wikipedia infobox, the first entity is presumed to be the one represented by the Wikipedia article containing that infobox. For example, the fact $f_{NumOfEmployee}$ = <*NumOfEmployee*(*Activision Blizzard, 5000*)*, 2011*> in the infobox in Fig. 1 is an outdated fact with respect to the fact $g_{NumOfEmployee}$ = <*NumOfEmployee*(*Activision Blizzard, 7300*)*, December 31 2011*> in the news pages in Fig. 2 about the number of employees of the mentioned company.

2.2 Processing Steps

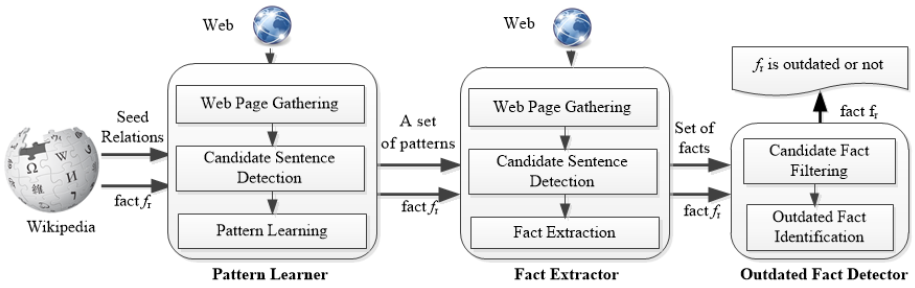


Fig. 3. The architecture of outdated Wikipedia infobox detection system

The architecture of our system to detect outdated information in Wikipedia infoboxes consists of three main components as shown in Fig. 3. The first component, *Pattern Learner,* uses the seed relations obtained from some Wikipedia infoboxes to search for a number of web pages that may contain candidate sentences related to a target relation, which is the relation of the possibly outdated fact to be detected. It then uses those sentences to learn patterns to extract instances of that relation. The second component, *Fact Extractor,* uses the learned patterns to extract those facts that are related to the target relation from the Web. The third component *Outdated Fact Detection* identifies if a fact is outdated or not by matching it with the candidate facts extracted from the Web.

2.3 Pattern Learner

The input of the pattern learning component is a small set of known instances of a target relation *r* that are automatically extracted from Wikipedia infoboxes. These relation instances are used as the seeds for the pattern learning process of the system. For example, the relation *NumberOfEmployee*(*Activision Blizzard, 5000*)*,* extracted from the infobox of company *Activision Blizzard* on Wikipedia in Fig. 1, can be used

as a seed for learning patterns related to the relation *NumberOfEmployee*. The pattern learning process has the three following steps.

Step 1: Web Page Gathering
From the information of seed instances of the target relation *r*, we use the Google search engine to gather web pages that may contain information of the seeds. The query keywords provided to the search engine are those words that represent the target relation, and the first and the second entities of the seed relation instances. Specifically, we manually build a gazetteer of keywords for each target relation. For each seed relation, the words representing the first entity are extracted from the URL of its corresponding Wikipedia article, while the words representing the second entity are those appear in the infobox containing that seed relation. Then, from the search results, only top-k web pages whose URLs or titles contain information about the first entity are collected.

For example, with the seed relation *NumberOfEmployee*(*Activision Blizzard, 5000*), the relation *NumberOfEmployee* is mapped to the keyword *"employ"*, using the constructed gazetteer. The URL of the Wikipedia article for company *Activision Blizzard* is *http://en.wikipedia.org/ wiki/Activision_Blizzard*. Therefore, the query keywords fed to the Google search engine are "*'Activision Blizzard' employ 5000*".

Step 2: Candidate Sentence Detection
From the search results in the above step, the text of each collected web page is extracted using library HtmlUnit[4]. Next, the text is segmented into a set of sentences using open natural language processing library OpenNLP[5]. Then, from this set, only candidate sentences that contain information about a seed relation are selected. Finally, each candidate sentence is classified into either the complete or incomplete-sentential form. A candidate sentence is considered to be a complete sentence if it has a verb phrase component. Otherwise, it is considered as an incomplete sentence. For example, from Figure 2, one has:

- s_1 = "*Employees: 7,300*" is an incomplete sentence, and
- s_2 = "*Activision Blizzard is headquartered in Santa Monica, California and employed about 7,300 people as of December 31, 2011*" is a complete one.

We also use library OpenNLP for POS tagging to classify sentences.

Step 3: Pattern Learning
Since there are two forms of candidate sentences, we need pattern structures suitable for each of the forms. Construction of patterns for the two forms is presented below.

Patterns for incomplete sentences: Given a seed relation $r(e_1, e_2)$ and an incomplete sentence s = "$w_1\ w_2 \ldots\ w_n$" where each w_i is a token, the pattern p for s is defined to be

[4] http://htmlunit.sourceforge.net/
[5] http://opennlp.apache.org/

of the form "t_1 t_2...t_n" where each t_i is a slot representing a token or a label related to the seed relation instance. Pattern p is built as follows:

- t_i is [r] if the keywords for r in the gazetteer is a substring of w_i.
- t_i is [`entity2`] if w_i represents the second entity of r.
- Otherwise, t_i is w_i.

For example, with the seed relation *NumberOfEmployees*(*Activision Blizzard, 5000*) and the incomplete sentence s_1 above, the pattern for s_1 is p_1 = "[`Number-OfEmployees`]: [`entity2`]".

Algorithm 1: Learning patterns

```
Input: S is a set of seeds on a target relation r
Output: P is a set of patterns
1: begin
2:    P ← {}
3:    for each seed s in S do begin
4:      keywords ← words representing the first entity of s +
5:                 words representing the relation of s +
6:                 words representing the second entity of s
7:      webPages ← getTop-k-WebPages(keywords)
8:    end for
9:    CS ← {} //CS is the set of candidate sentences
10:   for each web page w in webPages do begin
11:     C ← set of candidate sentences in w
12:     CS ← CS ∪ C
13:   end for
14:   for each candidate sentence c in CS do begin
15:     if (isCompleteSentence(c)) then
16:       p ← pattern of complete sentence c
17:     else
18:       p ← pattern of incomplete sentence c
19:     end if
20:     if (p not in P) then
21:       P ← P ∪ {p}
22:     end if
23:   end for
24:   return P
25: end
```

Fig. 4. Pattern learning algorithm

Patterns for complete sentences: The pattern construction is similar to that for incomplete sentences, with only one addition that, if w_i represents the first entity of the seed relation, then t_i is [`entity1`]. For example, with the seed relation *NumberOfEm-*

ployees(*Activision Blizzard, 5000*) and the complete sentence s_2 above, the pattern for s_2 is:

p_2 = "[entity1] *is headquartered in Santa Monica, California and* [Number-OfEmployees] *about* [entity2] *people as of December 31, 2011*".

Figure 4 presents our pattern learning algorithm. We note that duplicated patterns are removed in the resulting set of patterns, as shown in code lines 20 to 21 in the algorithm.

2.4 Fact Extractor

Patterns learned in the previous stage are used to extract facts from the Web that are related to the possibly outdated target fact in a Wikipedia infobox. By Definition 2.1, a fact consists of a relation instance and a time. The two following steps are to extract instances of the relation of the target fact. Identification of the associated time of an extracted fact is presented later in Section 2.5.

Algorithm 2: Extracting facts

```
Input:  s is a relation instance of a target relation
        P is a set of patterns
Output: F = {<f1, freq1>,...,<fn, freqn>} is a set of facts
        with their occurrence frequencies
1:  begin
2:    F ← {}
3:    keywords ←  words representing the first entity of s +
4:                words representing the relation of s
5:    webPages ←  getTop-k-WebPages(keywords)
6:    CS ← {} //CS is the set of candidate sentences
7:    for each web page w in webPages do begin
8:      C ← set of candidate sentences in w
9:      CS ← CS ∪ C
10:   end for
11:   for each candidate sentence c in CS do begin
12:     if (isMatchedWithPatterns(c, P)) then
13:       s ← extracted relation instance from c
14:       t ← extracted time of s
15:       freq ← extracted occurrence frequency of s
15:       F ← F ∪{(<s, t>, freq)}
16:     end if
17:   end for
18:   return F
19: end
```

Fig. 5. Fact extraction algorithm

Step 1: Web Page Gathering and Candidate Sentence Detection
This step is similar to the first two steps of the pattern learning stage. For the fact extraction stage, the difference is only that the target relation is used instead of a seed relation, and the second entity of the target relation is not used for searching related web pages. For candidate sentences, they need to contain only words representing the target relation and its first entity.

Step 2: Fact Extraction
Applying the learned patterns to the set of candidate sentences obtained from Step 1 above, a set of facts and their frequencies $\{(f_1, freq_1), ..., (f_n, freq_n)\}$ is extracted, where each f_i is a fact extracted from a candidate sentence that matches with some pattern, and $freq_i$ is the occurrence frequency of f_i in the set of web pages returned by the employed search engine. These frequencies are used to rank candidate facts for updating the target fact if it is outdated. For instance, pattern p_1 or pattern p_2 above may be applicable to extract the relation instance <*NumberOfEmployees*(*Activision Blizzard, 7300*)> from a certain related web page. The fact extraction algorithm is presented in Fig. 5.

2.5 Identification of Fact Time

Identification of the associated time of a fact depends on the source from which the fact is extracted. In this work, those sources are Wikipedia infoboxes and the Web. Time phrases in a text are recognized by pre-defined regular expressions.

Facts extracted from Wikipedia infoboxes
If a fact in a Wikipedia infobox contains a time as in Figure 1, then it is used as the associated time of the fact. Otherwise, the associated time of the fact is the time when the attribute value of the fact was added to the Wikipedia infobox. Such a time can be extracted from the updated history of a Wikipedia article. We use open library JWPL[6] (Java Wikipedia Library) to get this time information.

Fig. 6. A Google snippet returned for the query "Activision Blizzard employ"

Facts extracted from the Web
For a fact extracted from a sentence in a web page, if a time phrase is included in the sentence, then it is the associated time for the fact. Otherwise, the publication date of the web page is used as the associated time of the fact.

The publication date of a web page is the time when it was published on the web. We use the snippet of Google search engine for the web page to identify its publica-

[6] http://www.ukp.tu-darmstadt.de/software/jwpl/

tion date, which is a time phrase appearing at the beginning of the snippet, because Google can automatically identify it when crawling websites. For example, given the Google snippet in Fig. 6, the publication date of the corresponding web page is *29 Feb, 2012*. If the snippet does not contain a time phrase, the publication date is identified by analyzing the HTTP response from the employed web server for the corresponding web page.

2.6 Outdated Fact Detection

Given a target fact f_r from a Wikipedia infobox and a set $\{(f_{r1}, freq_1), ..., (f_{rn}, freq_{n,})\}$ of extracted facts from the Web with occurrence frequencies, f_r is considered to be outdated if and only if it is outdated with respect to some fact f_{ri} in that set, by Definition 2.2. The later fact for f_r with the highest occurrence frequency can be used to update f_r. Our outdated fact detection algorithm is Algorithm 3 presented in Figure 7.

Algorithm 3: Dectecting outdated facts

Input: f_r is a target fact to be checked whether it is outdated
$F = \{<f_{r1}, freq_1>,...,<f_{rn}, freq_n>\}$ is a set of extracted facts with occurrence frequencies
Output: a later fact to which f_r is outdated, or NULL otherwise.

```
1:  begin
2:    F* ← {} // subset of F
3:    for i from 1 to n do begin
4:      if (f_r.entity1 = f_ri.entity1 and f_r.entity2 ≠
5:          f_ri.entity2 and f_r.time < f_ri.time) then
6:        F* ← F* ∪ {<f_ri, freq_i>}
7:      end if
8:    end for
9:    if (isEmpty(F*)) then return NULL   end if
10:   f_r* = argmax_<f,freq>∈F* freq
11:   return f_r*
12: end
```

Fig. 7. Outdated fact detection algorithm

3 Experiments and Evaluation

To evaluate our proposed method in detecting outdated information in Wikipedia infoboxes, we have selected business companies with their numbers of employees shown in their infoboxes, and the target relation as the number of employees of a

company. Our constructed dataset includes 100 well-known companies in NASDAQ-100[7] index. For each company, we manually identified its latest number of employees from the company's official website, or from famous websites about jobs (e.g. glassdoor.com) or business (e.g. businessweek.com).

For pattern learning, seed relations are randomly chosen from the infoboxes of five companies in the dataset. For each seed relation, the top-100 related web pages from the results of the Google engine search are selected, from which patterns are extracted. After eliminating duplicated patterns, there are 7 patterns for extracting incomplete sentences and 15 patterns for extracting complete sentences. Then, for fact extraction for each company in the dataset using the learned patterns, the top-100 related web pages returned by the Google engine search are employed.

We use the accuracy measure based on the numbers of true positive (*TP*), false positive (*FP*), true negative (*TN*) and false negative (*FN*) cases, and calculated by the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

In our experiments, the meanings of the detected cases are as follows:

- *TP*: the number of employees of a company actually changed and the detection system says it changed.
- *FP*: the number of employees of a company actually changed but the system says it did not change.
- *TN*: the number of employees of a company actually did not change and the system says it did not change.
- *FN*: the number of employees of a company actually did not change but the system says it changed.

Table 1. The evaluation results on 100 companies

Cases	Result
TP	65
FP	17
FN	6
TN	12

Table 1 presents the obtained results with the number of each case. The accuracy of the proposed method is thus 77% (*TP* + *TN*). Besides, it shows that 82 (*TP* + *FP*) out of 100 companies did have their numbers of employees changed but there are no latest numbers in their Wikipedia infoboxes.

[7] http://en.wikipedia.org/wiki/NASDAQ-100

4 Conclusion and Future Work

In this paper we introduce the problem of automatically detecting outdated information in Wikipedia infoboxes. That is challenging and important because Wikipedia has grown fast and become a major information resource, but is still edited manually while the world is changing rapidly.

We have proposed an automatic method for detecting outdated facts in Wikipedia infoboxes with respect to the facts extracted on the Web. It is based on patterns automatically learned from the initiative information in Wikipedia infoboxes of the facts to be detected if being outdated or not. The method also suggests newer facts for updates based on their occurrence frequencies. The experimental results on one particular type of information, which is the number of employees of a company, show a good performance of the proposed method and reveal the striking truth about the outdated status of Wikipedia. The method is however general for arbitrary relations.

There are a number of possible ways to improve the proposed method. First, some machine learning techniques could be employed to find new attribute values in outdated Wikipedia infoboxes. Second, the credibility of an information resource could be taken into account, besides occurrence frequency, to rank and recommend correct and most up-to-date facts for revising Wikipedia. These are among the topics that we are currently working on in this area of research.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: Proceedings of International Semantic Web Conference, Korea. (2007)
2. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Elsevier Journal of Web Semantics (2008)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of The 2008 ACM SIGMOD International Conference on Management of Data, New York (2008)
4. Lange, D., Böhm, C., Naumann, F.: Extracting structured information from wikipedia articles to populate infoboxes. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, New York, ACM (2010)
5. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco (2001)
6. Liu, Q., Xu, K., Zhang, L., Wang, H., Yu, Y., Pan, Y.: Catriple: Extracting triples from wikipedia categories. In: Proceedings of The 3rd Asian Semantic Web Conference on The Semantic Web, Berlin (2008)
7. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge management, New York, ACM (2007)
8. Syed, Z., Saba, Finin, T.: Approaches for automatically enriching wikipedia. In: Collaboratively-Built Knowledge Sources and AI. Volume WS-10-02. (2010)

9. Syed, Z., Joshi, A.: Wikitology: Using wikipedia as an ontology. Technical report (2008)
10. Wu, F., Hoffmann, R., Weld, D.: Information extraction from wikipedia: Moving down the long tail. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York (2008)
11. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligent (2005)
12. Brin, S.: Extracting patterns and relations from the world wide web. In: International Workshop on The World Wide Web and Databases, London (1999)
13. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Conference on Digital Libraries, New York (2000)
14. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Communication ACM (2008)
15. Rozenfeld, B., Feldman, R.: Self-supervised relation extraction from the web. Knowledge Information System (2008)
16. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of International Joint Conference on Natural Language Processing of the AFNLP, Stroudsburg (2009)

Analyzing Variation Patterns In Quotes Over Time

Aurelien Lauf[1],[2], Mathieu Valette[1], and Leila Khouas[2]

[1] ERTIM (INALCO) – 49bis av. de la Belle Gabrielle, 75012 Paris
[2] AMI Software – Immeuble "Le Cristal", 1475 av. A. Einstein, 34000 Montpellier

Abstract. In the past few years, there has been a growing interest in detecting quotation families and in automatically analyzing transformation patterns. However, no work has provided a complete qualitative analysis of these variations. Through a comprehensive linguistic analysis, the goal of this paper is to study and categorize the way quotes from newspapers tend to evolve and deform over time. In order to help in observing patterns and variability, we apply global sequence alignment techniques, commonly used in bioinformatics. Recurrent patterns, such as the common deletion of words expressing modality, paraphrases, or strong synonymic variations, are listed and discussed. In addition to providing a better understanding of cultural dynamics in media, we believe that the categorization of variation regularities in quotes can help further enhance the quality of similar quotations clustering algorithms and other NLP tasks such as paraphrase identification.

Keywords: Discourse analysis, quotes, memes, multiple sequence alignment, linguistics, co-reference, modality

1 Introduction

Quoting is a very common practice, especially in journalism. A given quote can have many different versions: depending on the context, one can decide to focus on the beginning or the end of a quote, or on the contrary to write the whole segment. Quotes may also shrink or grow over time, for various reasons. Furthermore, one would usually assume that quotes are faithful to the original but this may not always be true: they are sometimes quoted from memory or willingly modified in order to convey a stronger meaning.

In the past few years, following [1] framework for "meme-tracking", there has been a growing interest in detecting quotation families and in automatically analyzing transformation patterns [2][3]. However, to our knowledge, no work has provided a complete qualitative analysis of these variations. The goal of this paper is, through a comprehensive linguistic analysis, to study and categorize the way quotes tend to evolve and deform over time. In order to help in observing patterns and variability, we apply global sequence alignment techniques, commonly used in bioinformatics. Recurrent patterns, such as the common deletion of words expressing modality, paraphrases, or strong synonymic variations, are

listed and discussed. In addition to providing a better understanding of cultural dynamics in media, we think that these variation regularities in quotes should be known in the NLP community; we believe that it can help further enhance the quality of similar quotations clustering algorithms and, to a lesser extent, paraphrase identification.

First, we will focus on previous works on this subject. We will then describe the textual data we used, and explain our approach. We will finally present and discuss the results of our analysis, focusing on the most interesting patterns.

2 Previous Works

In [1], authors present a clustering approach capable of identifying quotation families, i.e. all textual variants of each quotation, leading to the first large scale quantitative analysis of memes. In [2], hyperlinks between sources are added in order to adress the fidelity of information according to the type of source. Interestingly, authors show that most changes are introduced by media and that blogs are less likely to do so because they tend to simply copy and paste quotes. A new algorithm for quotation clustering is introduced in [3]. This method is based on [1] but relies on a linguistic approach. None of the above papers provide a comprehensive linguistic analysis to study and categorize the way quotes actually evolve and deform over time.

3 Description Of The Dataset

Our textual data is about the case relating to allegations of sexual assault against the former IMF managing director, during the year 2011. The documents have been collected using a metasearch engine with the following query: *dsk OR strauss-kahn OR strauss-khan*. The corpus comprised 27 439 news articles written in English.

We extract quotes from all documents, i.e. strings between quotation marks. We store for each of them the number of times they occur, the days they appear in (with the number of occurrences for each day) and the corresponding documents IDs in order to return to the text if needed: checking the context may be useful, e.g. to check whether two short quotes are really linked, or to understand why the journalist chose this particular segment of the full quote instead of another one. Using the approach described in [1], we produce quotes clusters which are groups of similar quotes allowing close textual variations. Using [4], quotes were lemmatized and stop words were filtered to help the clustering task[3]. In the remainder of this paper, we will talk about *quotes* instead of *quotes clusters*. Each *quote* has one or more *version(s)*. Table 1 shows an example of a *quote* which has 9 *versions*.

[3] Lemmatization and stop word filtering only occur during the quote clustering task. To avoid neutralizing differences, original quotes are used when analyzing variations.

Table 1. Illustration of a quote with all its 9 different versions, sorted by number of occurrences.

Version of the quote	**Occurrences**	**First and last dates**
"Offered a compelling and unwavering story"	66	05-20 to 06-06
"Compelling and unwavering story about what occured in the defendant room"	46	05-22 to 08-26
"Compelling and unwavering"	28	07-01 to 09-28
"The complainant in this case has offered a compelling and unwavering story about what occured in the defendant room"	3	05-24 to 07-03
"The victim has given a compelling and unwavering story about what happened in the defendant room"	2	05-20 only
"Offered compelling and unwavering story about what occurred in the defendant room"	2	05-24 only
"A compelling and unwavering story"	1	08-03 only
"Compelling and unwavering story"	1	08-03 only
"She offered a compelling and unwavering story"	1	08-23 only

Out of the 27 439 documents, our system has detected 22 099 quotes, many of them having only one version. We define the weight of a quote as the sum of the number of occurrences of each of its versions. For the present study, we decided to focus on the 100 best quotes according to this computed weight. In total, these 100 quotes represent 1039 different versions and 13 958 occurrences.

16% of quotes are translations, mostly from French. It is an interesting case because even though what was actually said was in a different language, it is reproduced between quotation marks. Quotation, which is strongly linked to intertextuality, is common and convenient in journalism to distantiate from what is being said (objectivity).

4 Description Of Our Approach

In order to observe patterns and variability, we perform global sequence alignment techniques commonly used in bioinformatics to identify similarities between sequences of DNA or protein [5]. These alignments are most of the time represented as rows within a matrix; similar elements are aligned on the same column and gaps may be introduced when an element has no match in the other sequences. Gap-to-gap matches are not allowed (a column cannot have only gaps). We are interested here in a word-to-word correspondence between all versions of a given quote instead of a residue-to-residue correspondence but the way to achieve it is similar [6]: the order of the words has to be preserved; when a word has no match in the other versions, a gap is introduced. Similar words are on the

same column. A simple example of perfectly aligned versions is shown in table 2.

Table 2. Illustration of the global alignment of the 9 versions of the quote shown in table 1. The last words are troncated in this example to stay on the same line. Dashes represent gaps. We can observe 2 replacements: *complainant/victim* and *offered/given*.

-	-	-	-	-	-	offered	a compelling and (...)
-	-	-	-	-	-	-	- compelling and (...)
-	-	-	-	-	-	-	- compelling and (...)
the	complainant	in	this	case	has	offered	a compelling and (...)
the	victim	-	-	-	has	given	a compelling and (...)
-	-	-	-	-	-	offered	- compelling and (...)
-	-	-	-	-	-	-	a compelling and (...)
-	-	-	-	-	-	-	- compelling and (...)
she	-	-	-	-	-	offered	a compelling and (...)

Using [7] dynamic programming algorithm, it is possible to easily get the best global alignment of two sequences. It is theoretically possible to generalize this method for more than two sequences using a hyper cube instead of a simple matrix. However, this generalization is unfeasible because it is known to be exponential in complexity. The only way to perform multiple sequences alignment is then to use heuristic methods which make locally optimal choices at each step, but are not guaranteed to find the optimal alignment [8][9]. The most used approach is the progressive alignment technique and the most famous implementation is ClustalW [10], which we use due to its efficiency.

The output alignment is noisy, especially when versions share only a few similarities. We thus had to manually correct all the mistakes. Nevertheless, it was a far less time consuming task than aligning all versions by hand. Results of our analysis are discussed in the following section.

5 Results And Discussion

Before discussing the observed variations and deletions between versions, we show here the results of some automatic quantitative analysis.

5.1 Diachronic Analysis

We store for each version of a quote the number of times it occurs, the days it appears in, and the number of its occurrences for each day. Using these data, we automatically calculate dispersion diagrams in order to check whether there is a correlation between the number of words of a version and the number of times it is reused. Figure 1 displays the relation between the number of words of a version and how many times it occurs. It also focuses on its lifespan, i.e. the

number of days it appears in the Press. Shorter versions seem to occur far more and have a better lifespan than long ones.

Furthermore, it is worth noticing that most of the versions appear right from the first day, i.e. there does not seem to be one original quote gradually modified through time.

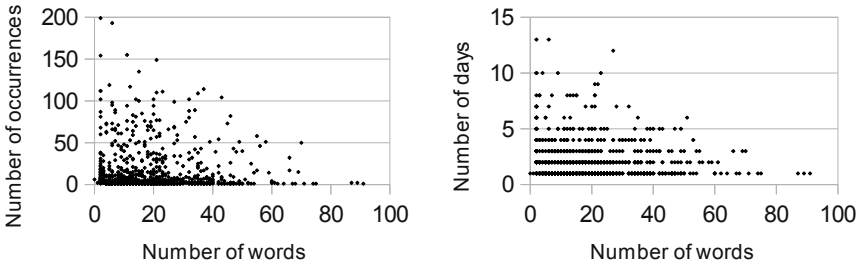


Fig. 1. Correlation between the number of words of a version and (left) the number of times it occurs, (right) the number of days it stays. The shorter a version is, the more and the longer it seems to be used.

5.2 Size And Variations

Next, using the sequence alignmnent technique described earlier, it is possible to count the number of deletions (gaps) and variations (replacements) within the different versions. Figure 2 displays the relation between the size of a quote (the number of words of its longest version) and the way it is altered. There is a tendency for longer quotes to be modified more than shorter ones, which is surprising because one would think that longer quotes are copy-pasted and not quoted from memory like shorter ones [2].

5.3 Convergence

We observed a tendency for quotes to converge to 2-3 words long phrases. About 60% of quotes staying more than one month are concerned (only 3% for shorter quotes). These phrases are what remains of the whole quote on the last day and are meaningful enough to recontextualize. Examples are shown in table 3.

5.4 Analyzing Variations

In order to better understand the way quotes are actually altered over time, we performed a comprehensive manual linguistic analysis of the variations and the deletions (see section 5.5 below) between versions.

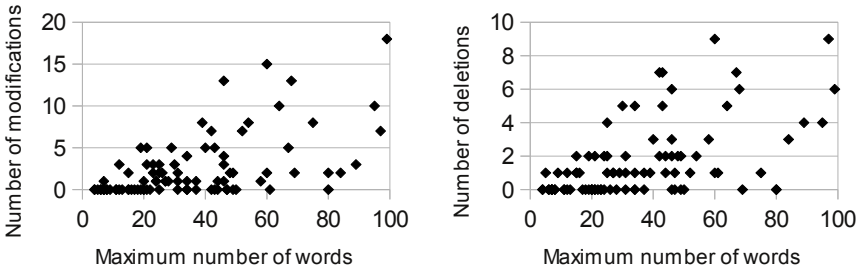


Fig. 2. Correlation between the number of words of the longest version of a quote (i.e. maximum number of words), and the way it is altered. Longer quotes tend to be less stable than short ones.

Table 3. Examples of what remains of the whole quote on the last day (3-4 months after the appearance of the quote).

He said he was leaving his IMF post with "infinite sadness" *so that he could devote full time to proving his innocence.*
That night he is in the custody of the New York Police Department facing the humiliating "perp walk".
How then, did she go from "compelling and unwavering" *to having her case dismissed due to lack of credibility?*

On our data set, 215 variations were observed. Table 4 shows the different types. We will focus on the three most important variations, i.e. synonymic variations, co-reference, and reformulation.

Synonymic variations are the most frequent type of modifications, mostly verbs and nouns. We noticed that translated quotes, which we discussed earlier, follow the same patterns than English ones, except for synonyms: about half (44%) of synonymic variations come from translated quotes. Furthermore, words from translations tend to have more variants, up to six:

- "It was a moral failing/failure/weakness/error/mistake/fault".
- "He tried to open/undo/remove my jeans".
- "It's important for a politician/man in politics/political man to be able to seduce".

Co-reference is a well known linguistic phenomenon: multiple elements (persons, actions or things) may have the same referent. Most of the time, later mentions of a previously introduced element are simpler, often reduced to pronouns. For example, consider the following version: "I'm rather proud of my husband reputation as a seducer". On some other versions, the "reputation as

Table 4. List of all the observed variation types, sorted by rate of occurrence. Most of the phenomenons can be subdivided. An example is given for each subtype.

Variation type	**Subtype**	**Example**
Synonyms (28.37%)	Verbs (40.98%)	*happen/occur*
	Nouns (37.70%)	*relationship/liaison*
	Adverbs (18.03%)	*gravely/seriously*
	Adjectives (3.28%)	*unjust/unfair*
Co-reference (16.74%)	Person: pronoun (41.67%)	*dsk/he*
	Person: reformulation (25%)	*woman/victim*
	Person: abbreviation (16.67)%	*district attorney/DA*
	Action (8.33%)	*what happened/it*
	Thing (8.33%)	*this incident/it*
Reformulation (12.09%)	Paraphrase (57.69%)	*has no idea/doesn't know*
	Syntactic variation (42.31%)	*a man with/this man has*
Spelling (11.16%)	(Common) misspelling (37.5%)	*whatsoever/what so ever*
	UK vs. US spelling (37.5%)	*honour/honor*
	Typo (25%)	*candidate/cadidate*
Determiners (8.84%)	Def. art./dem. (57.89%)	*the/this*
	Def. art./poss. (15.79%)	*the/my*
	Indef. art./dem. (10.53%)	*a/this*
	Indef. art./quantifier (5.26%)	*a/one*
	Def. art./indef. art. (5.26%)	*the/a*
	Dem./dem. (5.26%)	*this/that*
Conjugation (8.37%)	Tense (83.33%)	*have been/were*
	Person (11.11%)	*are/is*
	Mood (5.55%)	*put/puts*
Linking words (6.51%)	Prep./prep. (71.42%)	*on/in*
	Subord./subord. (21.43%)	*although/even if*
	Coord./coord. (7.14%)	*or/and*
Contractions (4.19%)	/	*was not/wasn't*
Number (1.86%)	/	*skill/skills*
Inversions (1.86%)	/	*still is/is still*

a seducer" is introduced out of the quote: *Anne Sinclair seemed forgiving of his reputed behavior. "No, I'm rather proud of it!" she told.* Sometimes, co-reference occurs for stylistic reasons to avoid repetitions of a word, e.g. "man" instead of "defendant". It may introduce some nuances though, willingly or not, e.g. "victim" does not carry the same implication than "complainant".

Reformulation can be divided into two subtypes : syntactic variations and paraphrases. The former includes many phenomenons, e.g. transition from direct to indirect speech or constituent order modification:

– "I just want to know if I need a lawyer" vs. "do I need a lawyer".

– "We expect him to be released tomorrow" vs. "we expect he will be released tomorrow".
– "Of which I am not proud" vs. "and I'm not proud of it".

Paraphrastic reformulation goes beyond simple syntactic modifications; whole phrases may be altered without modifying its global meaning:

– "I have no doubt" vs. "I am certain".
– "She has no idea what" vs. "she doesn't know what".
– "There were many reasons to believe" vs. "we continue to believe".
– "It was not just" vs. "it was more than".

5.5 Analyzing Deletions

129 deletions were observed. Table 5 displays the different types. We decided to ignore deletions at the beginning or the end of a version because they can be easily explained by indirect speech. We will focus on the most recurrent phenomenon: deletion of words expressing modality.

Modality [11] [12] is what allows speakers to express subjectivity. There is no consensus among researchers regarding categorisation of modality but most works agree on two main types: necessity and possibility. *Deontic modality* refers to permission and obligation (or moral desirability). *Alethic modality* is about (im)possibility and logical necessity. *Epistemic modality* indicates the speaker's judgment. Alethic and epistemic modality are often mixed because it might not be relevant to oppose what is logically true and what the speaker believes to be true [13]. We observe a strong tendency for words expressing epistemic modality to be omitted:

– "Forensic evidence (we believe) will not be consistent with a forcible account".
– "He is (obviously) not in a position to run the IMF".
– "(I think) it was a moral failing".

6 Conclusion

In this paper, we provided a comprehensive linguistic analysis and categorization of the variations in newspaper quotes over time. Quotes were semi-automatically aligned using a multiple sequence alignment technique, in order to help in detecting similarities, variations (replacements) and deletions (gaps).

We have detected some recurrent patterns, such as the common deletion of words expressing modality, paraphrases, or strong synonymic variations mostly among nouns and verbs. Furthermore, we highlighted an interesting tendency for quotes to converge to 2-3 words long phrases, powerful enough to summarize the whole context. We believe that a complete categorization of variation regularities

Table 5. List of all the observed deletion types, sorted by rate of occurrence. Deleted words are shown between parentheses.

Deletion type	**Subtype**	**Example**
Modality (20.15%)	Epistemic (80.77%)	*(I think) it is*
	Alethic (11.54%)	*(may) have*
	Deontic (3.85%)	*(have to) face*
	Affective (3.85%)	*I'm (sorry I'm) not*
Modifiers (18.60%)	Adjectives (50%)	*(physical) evidence*
	Adj. phrases (41.67%)	*influence (througout the world)*
	Noun adjuncts (8.33%)	*(selection) process*
Linking words (15.50%)	Coordinators (90%)	*my children (and) my friends*
	Conj. adverbs (10%)	*(indeed), we were intent on*
Determiners (10.85%)	Def. articles (50%)	*to (the) prosecutors*
	Possessives (28.57%)	*our guest and (our) staff*
	Indef. articles (21.43%)	*with (a) complete conviction*
Completives (10.07%)	Verbs (53.85%)	*I felt (that) I*
	Adjectives (30.77%)	*important (that) the*
	Nouns (15.38%)	*the idea (that) she*
Enumerations (7.75%)	/	*my strength (and all my energy)*
Time expressions (6.98%)	/	*I feel compelled (today) to*
Repetitions (5.43%)	Same referents (71.43%)	*this man (Mr. Strauss Kahn)*
	Same words (28.57%)	*a very (very) defensible case*
Intensity (4.65%)	/	*changed (a single) thing*

in quotes can help further enhance the quality of similar quotations clustering algorithms and other NLP tasks such as paraphrase identification.

Further studies have to be conducted on other sets of data though in order to validate our observations. Furthermore, we are aware that our results may be strongly influenced by the genre of our corpus made from news articles only; results should thus also be compared with different kind of news sources, i.e. blogs articles or tweets.

References

1. Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 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*, pages 497–506, Paris, 2009.
2. Matthew Simmons, Lada Adamic, and Eytan Adar. Memes online: extracted, subtracted, injected, and recollected. In *ICWSM 2011 - Proceedings of the 5th international AAAI conference on weblogs and social media*, Barcelona, 2011.
3. Elisa Omodei, Thierry Poibeau, and Jean-Philippe Cointet. Multi-level modeling of quotation families morphogenesis. In *Proceedings of the 2012 ASE/IEEE international conference on social computing (SocialCom 2012)*, Amsterdam, 2012.

4. Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, Manchester, 1994.
5. David Mount. *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press, 2004.
6. Robert Irving. Plagiarism and collusion detection using the Smith-Waterman algorithm. Technical report, Dept. of Computing Science, University of Glasgow, 2004.
7. Saul Needleman and Christian Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
8. Sing-Hoi Sze, Yue Lu, and Qingwu Yang. A polynomial time solvable formulation of multiple sequence alignment. *Journal of computational biology*, 13(2):309–319, 2006.
9. Cédric Notredame, Desmond Higgins, and Jaap Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.
10. Julie Thompson, Desmond Higgins, and Toby Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.
11. Paul Portner. *Modality*. Oxford University Press, 2009.
12. William Frawley, Erin Eschenroeder, Sarah Mills, and Thao Nguyen. *The expression of modality*. Mouton de Gruyter, 2006.
13. Frank Palmer. *Mood and modality*. Cambridge University Press, 1986.

Don't Use a Lot When Little Will Do: Genre Identification Using URLs

Pattisapu Nikhil Priyatam,[1] Srinivasan Iyengar,[2], Krish Perumal,[1] and Vasudeva Varma[1]

[1] Search and Information Extraction Lab, IIIT-Hyderabad, India
[2] Tata Research Development and Design Centre, TCS-Pune, India
nikhil.priyatam@research.iiit.ac.in,
venkatachary.srinivasaniyengar@tcs.com,
krish.perumal@research..iiit.ac.in, vv@iiit.ac.in

Abstract. The ever increasing data on world wide web calls for the use of vertical search engines. *Sandhan* is one such search engine which offers search in tourism and health genres in more than 10 different Indian languages. In this work we build a URL based genre identification module for *Sandhan*. A direct impact of this work is on building focused crawlers to gather Indian language content. We conduct experiments on tourism and health web pages in **Hindi** language. We experiment with three approaches - list based, naive Bayes and incremental naive Bayes. We evaluate our approaches against another web page classification algorithm built on the parsed text of manually labeled web pages. We find that incremental naive Bayes approach outperforms the other two. While doing our experiments we work with different features like words, n-grams and all grams. Using n-gram features we achieve classification accuracies of 0.858 and 0.873 for tourism and health genres respectively.

Keywords: Genre Identification, Focused Crawlers, Web Page Classification.

1 Introduction

The online content and consumption of Indian language web pages is increasing at a rapid pace[3]. This will only increase further as broadband and mobile-based Internet user base is expanding [1]. Thus, information retrieval problems such as Focused Crawling, Web Page Classification and Web Page Ranking need to be addressed for Indian languages. Even though these problems are addressed to a sufficient degree of satisfaction for English, a lot needs to be done when it comes to Indian languages. Efforts such as IndoWordNet [2] are trying to build excellent quality language resources. Still there is a clear scarcity of language resources for Indian languages. Many Indian language web pages use proprietary non-unicode fonts or non-standard character encodings. This is owing to the fact that many

[3] Vernacular Content Market in India, Dec 2008 : http://www.iamai.in/rsh_pay.aspx?rid=XsZAUn657BU=

web pages existed even before the unicode became a standard and thus have continued to use non-standard encodings. As building separate transcoders[4] for different non-standard fonts is effort intensive, most of the content of these web pages is rendered useless for many information retrieval tasks. In Information Retrieval, Web Page Classification (WPC) has many significant applications and is especially pivotal in building genre specific search engines. *Sandhan*[5] is one such search engine which offers search for tourism and health genres across 10 different Indian languages. In this work, we describe a light-weight, genre specific web page classification algorithm for Hindi. Our approach looks at clues in the URL of a web page to decide its genre. We test our approach on two genres - tourism and health. As a byproduct, we have other added advantages that come with this approach. These include classification speed and economy of operation in terms of bandwidth, processing and storage.

The flow of this paper is as follows. We discuss relevant literature on the use of URLs for different IR activities in section 2. In section 3 we describe our experimental dataset. In section 4 we describe our system architecture. We propose different approaches for WPC using URLs in section 5. Section 6 describes our experimental setup. In section 7 we present the metrics used for evaluation. In section 8 we report our results when evaluated against a WPC algorithm [3]. We conclude our discussion with future directions in section 9.

2 Literature Survey

URL based methods have several advantages and they should be employed when:

- Classification speed must be high.
- Content filtering is needed before an objectionable or advertisement page is downloaded.
- Page's content is hidden in images or **non-standard encodings**.
- Annotation needs to be performed on hyperlinks in a personalized web browser, without fetching the target page.
- Focused crawler wants to infer the topic of a target page before devoting bandwidth to download it.
- Language of the page needs to be identified.

Baykan et al. [4] describe an approach to classify web pages into 15 different topics such as sports, news, adult, shopping, etc. for English language pages. Their feature list consists of topic specific words and their n-grams. They train separate binary classifiers (SVM) for each topic. Although they use words from the first two levels of the ODP hierarchy for a particular topic, some manual post processing was done to remove non-topic-specific words such as "online" and "games".

Baykan et al. [5] describe a method to identify the language of the web page by analyzing the URL. They apply a variety of machine learning algorithms

[4] A transcoder converts non-unicode text into unicode
[5] http://www.clia.iitb.ac.in:8080/sandhan-prsg/locale.jsp?en

to the language identification task and evaluate their performance in extensive experiments for five languages: English, French, German, Spanish and Italian.

Kan et al. [6] introduce features such as position, length and sequence of tokens in a URL for the task of web page classification.

Shih et al. [7] propose new features and algorithms for automating web page classification tasks such as content recommendation and ad blocking. Apart from analyzing the URL tokens, they study the placement of these links in the referring page i.e. its HTML tree structure. They develop a machine learning model and algorithm using such features.

Kan et al. [8] present a couple of techniques to segment and expand URLs into tokens for performing effective web page categorization. Initial baseline segmentation is done using non-alphanumeric characters, uppercase-lowercase transitions and digits. Further segmentation of individual chunks is done using the information content of every possible partition. The probabilities required for this information theory based method were estimated by observing document frequency of tokens over several million web pages. The chunks are further split and expanded based on previously seen web page titles using a weighted non-deterministic finite-state transducer.

Hernndez et al. [9] mention an unsupervised method to build URL patterns for web page clustering.

Anastácio et al. [10] talk about categorizing documents according to their implicit locational relevance. One of the features they use is URL n-grams. They vary n between 4 and 8 and assign weights according to the TF-IDF scheme.

Ma et al. [11] use lexical and host-based features of URLs to identify spam sites. Further they use online learning algorithms to classify malicious URLs with high accuracy. Toyoda et al. [12] further classify them into different spam topics. Kołcz et al. [13] use web-graph information along with the basic URL-based approach and show improved results for classifying webpages.

Abramson et al. [14] give an exhaustive list of features that can be extracted from a URL, which include syntactic style features, semantic style features, part-of-speech (POS) tags, punctuations and special characters. They build classifiers (Naive Bayes and SVM) using *Santini* and *Syr7* datasets.

Charu et al. [15] provide a framework for intelligent crawling where the crawler gradually learns the linkage structure as it progresses. Their crawler incrementally learns the tokens in the URL which are useful for classification.

Even though genre identification from the URL of a web page is an important problem and plays a key role in solving Information Retrieval problems like focused crawling and WPC, there is no single work reported for Indian languages. Most of the existing approaches require huge amount of resources such as heavy training data, already existing corpus or web graph information collected from a search engine [14] [13]. One cannot always afford to have these resources. Many works like [14] report their results on toy datasets and hence are not scalable to be used in search engines like *Sandhan*[6]. In this work we build a URL based genre identification system on a huge dataset using minimal resources and training.

[6] http://www.clia.iitb.ac.in:8080/sandhan-prsg/locale.jsp?en

3 Experimental Dataset

To start with, we have a set of approximately 3000 seed URLs collected and tagged manually into tourism, health and miscellaneous genres as shown in table 1. With these seed URLs we crawl the web till a depth of 3. After eliminating other language pages and pages which are not properly parsed (about 25000 in number), we are left with 94995 **Hindi** pages.

Since such a huge number of pages cannot be tagged manually, we use the

Table 1. Statistics of data collected

Genre	No of Web Pages
Tourism	885
Health	978
Miscellaneous	1354

WPC algorithm based on the parsed text of the web pages as used in [3]. The web pages corresponding to the URLs mentioned in table 1 are used to train this algorithm. The 10 fold cross validation accuracy of this algorithm is 81.89% . Using this we tag the 94995 Hindi web pages. This is **assumed** to be our golden data and is used to evaluate our approach. Note that in our golden data we have preserved the order in which the pages were crawled. An interesting observation is that more than 25% of web pages are not used because of parsing issues.

4 System Architecture

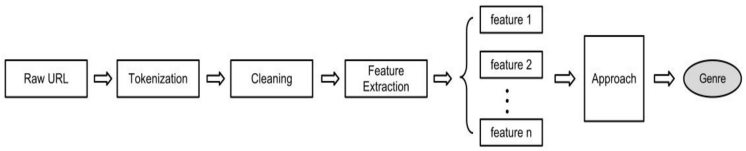


Fig. 1. System Architecture

Figure 1 describes the overall system architecture. A given URL is initially tokenized using common separators like full stop, hyphen, etc. The tokens which do not contain genre specific information are removed in the cleaning phase. In the feature extraction module various features like words and n-grams (4 to 8 grams) are extracted as used by Anastácio et al. [10]. Finally, the approach consumes these features to output the genre of the URL. The following example explains the system architecture in brief.

Raw URL: http://origin-hindi.webdunia.com/tourism.html
Tokenization: http origin hindi webdunia com tourism html
Cleaning: origin hindi webdunia tourism
Feature Extraction: Words - {origin, hindi, webdunia, tourism}
4-Grams - {orig, rigi, igin, ginh, inhi, nhin, hind, indi,.....}
5-Grams - {origi, rigin, ginhi, inhin, nhind, hindi,.....}
6-Grams - {origin, riginh, iginhi, ginhin, inhind, nhindi,....}
7-Grams - {originh, riginhi, iginhin, ginhind, inhindi,.....}
8-Grams - {originhi, riginhin, iginhind, ginhindi,.....}
All-Grams - Set of all grams (4 to 8) mentioned above

In the following section we propose three approaches to solve the problem of identifying the genre of a web page using only its URL, namely list based, naive Bayes and incremental naive Bayes.

5 Proposed Approaches

5.1 List Based Approach

This is a well known method which uses a list of tokens/words that are considered to be indicative of the genre. The list is matched against the tokens of the URL under consideration. For the URL to belong to that genre, at least one of its tokens should be present in the list. We now describe methods to prepare such a list for Hindi.

Manual Collection: A list of words (List 1) belonging to a specific genre are manually collected. For example, the list for tourism genre will contain words like travel, tour, flight, etc. and the list for health genre will contain words like health, disease, care, vitamin etc. in health genre. A glaringly obvious disadvantage of this approach is that it is effort intensive and requires genre/domain expertise.

Via an External Corpus: A corpus of sentences belonging to tourism and health genres are publicly available [7] for English and various Indian languages.

Figure 2 part (b-1) depicts the creation of the list using the genre specific corpus of English sentences. Nouns are extracted from the English sentences to give rise to List 2. This list is further augmented using WordNet which gives rise to List 3.

It is important to note that URL tokens for Hindi web pages might contain Hindi words transliterated to English. This can not be captured by the earlier lists. So we use a corpus of Hindi sentences. Figure 2 part (b-2) shows the creation of this list. Nouns are extracted from the Hindi sentences and transliterated to

[7] `www.tdil-dc.in/index.php?option=com_up-download&view=download&lang=en&limitstart=70`

English using a proprietary transliteration engine. This gives rise to List 4. This list is further augmented using IndoWordNet which gives rise to List 5.

Via a Retrieval System: The previous approach depends on the presence of a tagged sentence corpus specific to a particular genre. This is a severe drawback since such a corpus may not be available for many genres. To overcome this limitation, we explore the use of the web to get a list of words specific to almost any genre. We propose the use of Wikipedia categories and search engines for this purpose. Wikipedia is an online collaborative knowledge sharing system. All pages in Wikipedia are assigned to at least one category. Each of these may be subdivided into several subcategories. Since Wikipedia content is constantly updated by a diverse audience, almost every genre will have its representation in Wikipedia as a category. We use subcategory information because of the wide coverage and diversity it offers within the genre.

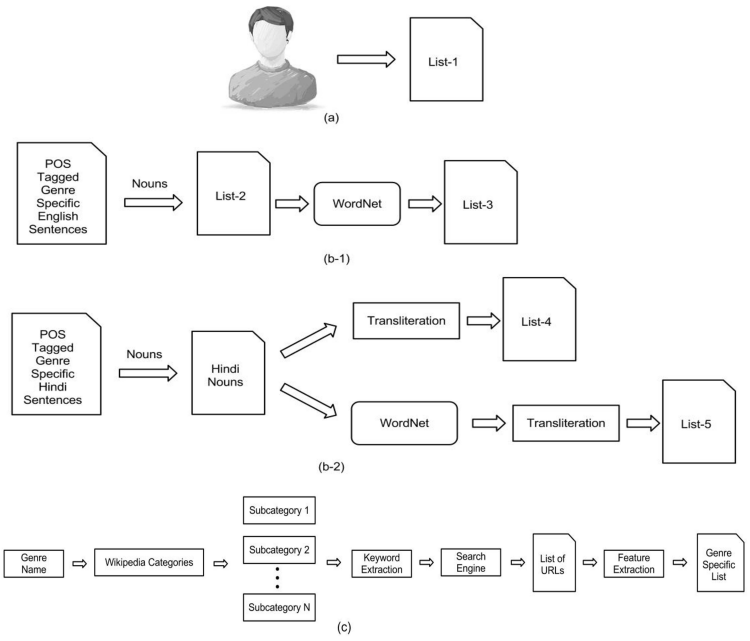


Fig. 2. List Based Approaches (a) Manual Collection (b-1) Via an External English corpus (b-2) Via an External Hindi corpus (c) Via a Retrieval System

Figure 2 part (c) shows how a genre specific list is created via a retrieval system. Given a genre, the corresponding Wikipedia category and all of its subcategories are extracted. Keywords are manually crafted for each of these subcategories and queried on to a search engine. Features are extracted from the top 'k' URLs to get the genre specific list of tokens.

There are several disadvantages of using the list based approach. Firstly, it is blind to the number of matching tokens in the URL. Moreover, one might not be able to capture all the genre specific tokens exhaustively. Further, attempts at increasing the list size to include more genre specific tokens may introduce noise, that is, generic words or words specific to other genres might creep in.

5.2 Naive Bayes Approach

In this method we train a naive Bayes algorithm to learn the patterns in URLs of a specific genre. Naive Bayes is the most popular text classification algorithm and is known to be accurate despite the independence assumption that it makes. Also, the training time of naive Bayes is much less when compared to other machine learning algorithms.

5.3 Incremental Naive Bayes

The naive Bayes approach is limited by the amount of training data. Moreover, the prior probabilities used in the algorithm depend on the distribution of previously observed training data. This introduces a classification bias towards the dominant class. To counter both these, we use an incremental version of the naive Bayes algorithm along with uniform prior probabilities. This has several advantages. This method incorporates the knowledge from all previously observed URLs and constantly improves as more and more URLs are observed. The uniform prior probabilities ensure that the current classification task is not affected by previous class distribution of training data.

Specifically in the context of focused crawling, where URL based genre identification is of significant use, the incremental naive Bayes algorithm is an ideal approach. In such a scenario, this algorithm can continue to train on newly crawled URLs and improve its accuracy. All the previous approaches would be unable to take advantage of the information from these newly crawled URLs.

6 Experimental Setup

In this work, we experiment with Hindi language tourism and health URLs. For the manual list based approach, a total of 29 words were selected from each genre. In case of the list based approach via a retrieval system, tokens of less statistical significance were not considered to be representative of the genre, and hence, eliminated. Similarly, tokens like wiki, pedia, etc. which were found in both genres were also removed. In the naive Bayes approach, we use 60% of the annotated data for training and report the results for the remaining 40% testing data.

7 Evaluation Metrics

We evaluate our results using different metrics like classification accuracy and rate of change of classification accuracy wherever applicable.

$$Classification\ Accuracy = \frac{\#\ of\ correctly\ classified\ samples}{Total\ \#\ of\ samples} \quad (1)$$

8 Results and Analysis

Table 2. Accuracy of List Based Approach

Genre	Tourism	Health
List 1	0.766	0.753
List 2	0.359	0.284
List 3	0.339	0.276
List 4	0.589	0.331
List 5	0.362	0.270

Table 3. Accuracy of List Based Via a Retrieval System

Genre	Tourism	Health
4 grams	0.399	0.344
5 grams	0.483	0.477
6 grams	0.575	0.562
7 grams	0.632	0.607
8 grams	0.657	0.633

Table 4. Accuracy of Naive Bayes

Genre	Tourism	Health
Words	0.848	0.446
4 grams	0.845	0.363
5 grams	0.845	0.463
6 grams	0.844	0.433
7 grams	0.839	0.456
8 grams	0.846	0.509

Table 2 shows the results for the list based approach using manual collection and external corpus respectively. Considering that a certain amount of manual intelligence was used, these results are not satisfactory, especially for the health genre. There may be several reasons for this. While creating these lists, n-grams were not used. The manual way of collecting cannot be exhaustive, while the

Table 5. Accuracy of Incremental Naive Bayes Approach

Genre	Tourism	Health
Words	0.849	0.862
4 grams	0.849	0.851
5 grams	0.858	0.865
6 grams	0.860	0.873
7 grams	0.860	0.875
8 grams	**0.858**	**0.873**
All grams	0.856	0.867

corpus based approach cannot be accurate due to the presence of noise. Both of them are incapable of capturing URL vocabulary. On the other hand, the retrieval based system uses n-gram features and captures URL vocabulary. As shown in table 3, it outperforms both manual and corpus based approaches.

From table 4, we can see that even after giving 60% of the labeled set as training data, the naive Bayes approach does not perform satisfactorily on the health genre. Even though this system performs well on the tourism genre, it is not scalable because it uses an enormous amount of data for training. Moreover, if the amount of training data is less, the performance is sure to reduce. For the naive Bayes to perform well, the training and testing data must come from the same distribution. It is highly unlikely that the fixed amount of training data that it gets is a good representation of the entire genre.

The results for the incremental naive Bayes approach are shown in table 5. While doing experiments we use various feature spaces which include words, n-grams and all grams. We have seen that for this particular problem, n-gram features perform better than words. This is due to the fact that URLs are generally noisy, and using n-grams handles noise, spell variations and short forms in a much more efficient way.

The rate of change of accuracy curves for table 5 are shown in figures 3 and 4 for tourism and health genres respectively.

Note that Y-axis shows the classification accuracy which varies between 0 and 1. But we have showed only the variations between 0.83 to 0.89. This is because the learning curve is very fast for the first 100 URLs, and if we try to show the classification accuracy between 0 and 1, the graph becomes skewed. We can see that 5 out of 6 n-gram features work better than word features.

9 Conclusion and Future Work

In this work we have built a web page classification algorithm for Hindi tourism and health genres using only URLs of the web pages. All the experiments were done on a huge dataset containing thousands of pages. We have specifically shown its usage in the context of a focused crawler. To the best of our knowledge, there is no prior work reported for this problem in Indian languages. We are the first ones to provide a working solution for this. We propose three different kinds of

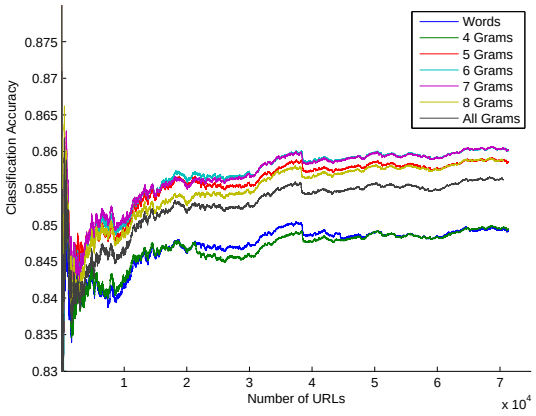


Fig. 3. Performance comparison of different features in Tourism genre

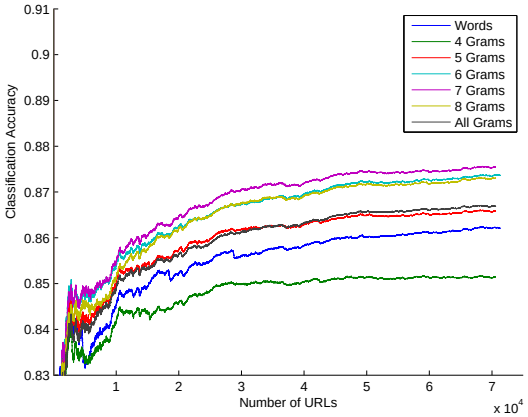


Fig. 4. Performance comparison of different features in Health genre

approaches, from the very basic to advanced, to solve this problem: list based, naive Bayes and incremental naive Bayes approaches. The incremental naive Bayes gave the best results. Our proposed approaches are generic and can be easily extended to other languages and genres.

In future we want to try a hybrid approach which uses all the three approaches proposed in this work. We also want to conduct experiments for various other Indian languages and extend it to other genres like sports, entertainment etc.

References

1. Sharma, A., Thoppil, D.: Google sees india web explosion (2012)
2. Bhattacharyya, P.: Indowordnet. In: LREC 2010. (2010)
3. Priyatam, P., Vaddepally, S., Varma, V.: Domain specific search in indian languages. In: Proceedings of the first workshop on Information and knowledge management for developing region, ACM (2012) 23–30
4. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely url-based topic classification. In: Proceedings of the 18th international conference on World wide web, ACM (2009) 1109–1110
5. Baykan, E., Henzinger, M., Weber, I.: Web page language identification based on urls. Proceedings of the VLDB Endowment **1**(1) (2008) 176–187
6. Kan, M., Thi, H.: Fast webpage classification using url features. In: Proceedings of the 14th ACM international conference on Information and knowledge management, ACM (2005) 325–326
7. Shih, L., Karger, D.: Using urls and table layout for web classification tasks. In: Proceedings of the 13th international conference on World Wide Web, ACM (2004) 193–202
8. Kan, M.: Web page classification without the web page. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM (2004) 262–263
9. Hernndez, I., Rivero, C.R, Ruiz, D., Corchuelo, R.: A statistical approach to url-based web page clustering. In: Proceedings of the international World Wide Web conference, ACM (2012)
10. Anastácio, I., Martins, B., Calado, P.: Classifying documents according to locational relevance. Progress in Artificial Intelligence (2009) 598–609
11. Ma, J., Saul, L., Savage, S., Voelker, G.: Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM (2009) 681–688
12. Toyoda, Y., Kitsuregawa, M.: Topic classification of spam host based on urls. In: DEIM 2010. (2010)
13. Kołcz, A., Hulten, G., Szymanski, J.: Topical host reputation for lightweight url classification. Technical Report MSR-TR-2012-67, Microsoft (2012)
14. Abramson, M., Aha, D.: Whats in a url? genre classification from urls. In: Proceedings of Association for the Advancement of Artificial Intelligence (www.aaai.org), AAAI (2012)
15. Charu, C., Al-Garawi, F., Yu, P.: Intelligent crawling on the world wide web with arbitrary predicates. In: Conference proceedings of World Wide Web 2010. (2010)

Extracción automática de un diccionario de colocaciones en español

Sulema Torres-Ramos

Centro de Investigación en Computación (CIC-IPN),
Unidad Profesional Adolfo López Mateos,
Av. Juan de Dios Bátiz s/n esquina M. Othón de Mendizábal,
Zacatenco, México, D.F. 07738, México.
sulema7@gmail.com

Resumen. Las colocaciones son pares de palabras de contenido que forman las relaciones sintácticas de dependencia razonables, directamente o a través de palabras funcionales. Tales pares tienden usarse en los textos más frecuentemente de lo esperado por casualidad. El texto en lenguaje natural consiste casi totalmente de tales colocaciones. La información de las palabras que forman colocaciones es útil en diferentes aplicaciones de procesamiento de lenguaje natural. Actualmente existen corpus etiquetados con estructura sintáctica mismos que pueden ser útiles para la extracción de colocaciones.

En este artículo se presenta la extracción automática de un diccionario estadístico grande de colocaciones a partir de un corpus con las estructuras sintácticas marcadas manualmente. Las relaciones de dependencias encontradas en tal corpus, junto con sus frecuencias, constituyen nuestro diccionario de colocaciones.

Palabras clave: Lingüística Computacional, Procesamiento de Lenguaje Natural, Colocaciones, Ambigüedad Sintáctica.

1 Introducción

Una colocación es la relación entre dos palabras o un grupo de palabras que frecuentemente se usan de manera conjunta formando una expresión común [1]. Algunos ejemplos de colocaciones en español son *sistema político*, *seguro de vida*, *núcleo familiar*, etc.

Supongamos a un estudiante escribiendo un ensayo acerca del medio ambiente. Él conoce los temas que desea cubrir y tiene las ideas y los argumentos para hacerse entender. Además posee un repertorio de vocabulario útil, especialmente de sustantivos de alto contenido como *medio ambiente*, *contaminación*, *capa de ozono*. Lo que hace falta son las palabras que pueden ligar el vocabulario de alto contenido para convertirlo en un texto coherente -como un argumento o una narración-. La contaminación es un problema, pero ¿qué se necesita hacer al respecto? Buscando en un diccionario y examinando rápidamente en la sección de verbos nos arrojara las opciones de *evitar* / *prevenir* / *combatir* / *controlar* / *pelear* / *limitar* / *minimizar* / *reducir* /

monitorear. Con la ayuda de un diccionario común el estudiante puede escoger entre las opciones, la que exprese mejor lo que quiere decir. Sin embargo, si las colocaciones son difíciles de producir para un hablante no nativo [2,3], mucho más difícil lo es para una computadora.

Muchos esfuerzos se han hecho por generar manualmente diccionarios de colocaciones, sin embargo el tiempo y costo de esta tarea es muy alto. Actualmente existen corpus etiquetados con estructuras sintácticas que son la base para la extracción de colocaciones; de ahí surge la necesidad y relevancia de la obtención automática de colocaciones, utilizando los recursos existentes.

En este artículo se presenta un método automático para la extracción de colocaciones basado en un corpus etiquetado con las estructuras sintácticas. Específicamente, se llevó a cabo la extracción de un diccionario de colocaciones en español a partir del corpus etiquetado en español Cast3LB.

El artículo se organiza como sigue: primero, describimos a detalle las colocaciones, sus tipos y principales aplicaciones en lingüística computacional (sección 2) y se presentan los dos principales formalismos sintácticos en la sección 3. Después se presenta el método utilizado para la extracción del diccionario de colocaciones (sección 4). En la sección 5 se presenta la evaluación y resultados obtenidos. Al final se presentan las conclusiones y trabajo futuro.

2 Colocaciones

Hay mucha discusión y trabajo relacionado sobre colocaciones [4,5,6]. Dependiendo de los intereses y puntos de vista, los investigadores se enfocan en diferentes aspectos de las colocaciones.

Una de las definiciones más entendibles y usadas se encuentra en el trabajo lexicográfico presentado en [8]. La definición es la siguiente: una colocación es una combinación recurrente y arbitraria de palabras.

En [9] se define una colocación como: un par de palabras de contenido conectadas sintácticamente y que tienen compatibilidad semántica. Por ejemplo: *tomar* una *decisión*, *escuchar* la *radio*, *tocar* la *guitarra*, etc., donde los componentes de la colocación (colocativas) están subrayados.

Por palabras de contenido entendemos aquellas que tienen un significado, entre ellas se encuentran los sujetos, verbos, adjetivos, por ejemplo: perro, niño, comer, bonita, etc. Las palabras que no tienen contenido son los artículos como las, el, esos, etc.

La conexión sintáctica es entendida en las gramáticas de dependencias y ésta no es precisamente la coocurrencia de colocativas en un intervalo pequeño de texto. La colocativa rectora gobierna sintácticamente a la colocativa dependiente, estando adjunta a ella directamente o mediante una palabra auxiliar (usualmente una preposición). Secuencialmente, las colocativas pueden estar a cualquier distancia una de otra en una oración, mientras que en un árbol dependencias están muy cercanas.

2.1 Propiedades de las colocaciones

En esta sección, presentamos cuatro propiedades de las colocaciones que tienen relevancia en aplicaciones de lingüística computacional.

Las colocaciones son arbitrarias

Las colocaciones son difíciles de producir para un hablante no nativo [3]. No se trata simplemente de traducir palabra por palabra (word-for-word) lo que le gustaría al hablante decir en su lengua nativa. La tabla 1 muestra que la traducción palabra por palabra de "*to see the door*" corresponde en ambas direcciones de los cuatro lenguajes diferentes. Al contrario, traducir palabra por palabra la expresión "*to break down/force the door*" no tiene correspondencia en ambas direcciones en ninguno de los lenguajes.

La coocurrencia de "*door*" y "*see*" es una combinación libre, mientras que la combinación de "*door*" y "*break down*" es una colocación.

Para los hablantes no nativos de inglés es difícil construir correctamente la frase *"to break down a door"*.

Tabla 1. Comparaciones lingüísticas cruzadas de colocaciones

Lenguaje	Inglés	Traducción	Correspondencia en inglés
Francés	to see the door	voir la porte	To see the door
Alemán	to see the door	die Tür sehen	To see the door
Italiano	to see the door	vedere la porta	To see the door
Español	to see the door	ver la puerta	To see the door
Francés	to break down/force the door	enfoncer la porte	to push the door through
Alemán	to break down/force the door	die Tür aufbrechen	to break the door
Italiano	to break down/force the door	sfondare la porta	to hit/demolish the door
Español	to break down/force the door	tumbar la puerta	To fall the door

Traducir de un lenguaje a otro requiere más que buen conocimiento de estructura sintáctica y representación semántica, porque las colocaciones son arbitrarias, y deben ser fácilmente disponibles en ambos idiomas para que la traducción automática sea eficiente.

Las colocaciones son dependientes del dominio

Además de las colocaciones no técnicas tales como las que se presentaron antes, las colocaciones específicas del dominio son numerosas. Éstas son a menudo totalmente inentendibles para alguien ajeno al dominio. Contienen una gran cantidad de términos técnicos. Además, las palabras comunes se utilizan diferentemente. En el dominio de la navegación [10], por ejemplo, algunas palabras son desconocidas al lector no-familiar; la horca, y el sotavento son totalmente sin sentido para alguien ajeno a este dominio. Algunas otras combinaciones no contienen al parecer ninguna palabra técni-

ca, pero estas palabras adquieren un significado totalmente diferente en el dominio. Por ejemplo, un *traje seco* no es solamente un traje que está seco sino un tipo especial de traje usado por los marineros para permanecer seco en condiciones atmosféricas difíciles.

Dominar lingüísticamente un área específica requiere más que un glosario, requiere conocimiento de colocaciones dependientes del dominio.

Las colocaciones son recurrentes

La propiedad recurrente significa que las combinaciones de palabras no son excepciones, sino que se encuentran frecuentemente repetidas en un contexto dado.

Combinaciones de palabras como "tomar una decisión", "hacer un favor" son típicas del lenguaje, y colocaciones como "juntar hilos" son características de dominios específicos. Ambos tipos son frecuentemente usados en contextos específicos.

Las colocaciones son conjuntos de cohesión léxica

Por cohesión léxica [11] se entiende que la presencia de una o varias palabras de la colocación frecuentemente implica o sugiere el resto de la colocación. Esta propiedad es la más usada por lexicógrafos cuando compilan colocaciones [12,13].

Los lexicógrafos usan el juicio lingüístico de la gente para decir cuales son colocaciones y cuales no [14]. Ellos aplican cuestionarios a la gente, como el que se muestra en la siguiente tabla.

Tabla 2. Prueba *llenar-el-espacio* de Benson [8]

Oración	Candidatos
If a fire breaks out, the alarm will ??	ring/go off/ sound/start
The boy doesn't know how to ?? his bicycle	drive/ride/conduct
The American congress can ?? a presidential veto	ban/cancel/delete/reject
Before eating your bag of microwavable popcorn, you have to ?? it	cook/nuke/broil/fry/bake

Este cuestionario contiene las oraciones usadas por Benson para compilar el conocimiento de colocaciones para el diccionario BBI [15]. Cada oración tiene una ranura en blanco que puede ser fácilmente llenado por un hablante nativo (en este caso de inglés). En cambio, un hablante no nativo de inglés no encontraría las palabras faltantes automáticamente, sino que consideraría la lista de opciones de las palabras que tienen las características semánticas y sintácticas apropiadas, tales como las que están dadas en la segunda columna.

Como consecuencia, las colocaciones tienen una distribución estadística particular [5,16]. Esto significa que la probabilidad de que cualesquiera dos palabras adyacentes, por ejemplo, "arenque rojo" es considerablemente mayor que la suma de probabilidades de "rojo" y "arenque". Las palabras no pueden ser consideradas como variables independientes.

2.2 Tipos de colocaciones

Las colocaciones vienen en una gran variedad de formas. El número de palabras implicadas así como la forma de implicarlas puede variar mucho. Algunas colocaciones son muy rígidas, mientras otras son muy flexibles. Por ejemplo, una colocación compuesta por "tomar" y "decisión" puede aparecer como "tomar una decisión", "decisiones por tomar", "tomar una gran decisión", etc. En cambio, una colocación como "agente de ventas" puede aparecer sólo de una forma; esta es una colocación muy rígida, una expresión fija.

Se identifican tres tipos de colocaciones [11]: oraciones nominales rígidas, relaciones predicativas y plantillas de frase. A continuación se explican cada una de ellas.

Relaciones predicativas

Una relación predicativa consiste en dos palabras que se usan juntas repetidamente en una relación sintáctica similar [11]. Este tipo de colocación es la más flexible.

Por ejemplo, un sustantivo y un verbo formarán una relación predicativa si se usan juntos en varias ocasiones con el sustantivo como el objeto del verbo, "tomar-decisión" es un buen ejemplo de una relación predicativa. Así mismo, un adjetivo que frecuentemente modifica un sustantivo, como "niño-pequeño", es también una relación predicativa.

Esta clase de colocaciones se relaciona con las funciones léxicas de Mel'cuk [6,7], y las relaciones tipo L de Benson [15].

Oraciones nominales rígidas

Esta clase de colocaciones envuelve secuencias ininterrumpidas de palabras como "bolsa de valores", "procesamiento de lenguaje". Estas pueden incluir sustantivos y adjetivos, así como palabras de clase cerrada, y son similares al tipo de colocaciones recuperadas por [17,18]. Son el tipo más rígido de colocaciones. Algunos ejemplos son, "producto interno bruto", "impuesto al valor agregado", etc.

En general, las oraciones nominales rígidas no se pueden descomponer en fragmentos más pequeños sin perder su significado; son unidades léxicas en y de sí mismas. Por otra parte, frecuentemente se refieren a conceptos importantes en un dominio específico, y varias oraciones nominales rígidas se pueden utilizar para expresar el mismo concepto.

Plantillas de frase

Consisten en frases idiomáticas que contienen una, varias o ninguna ranura en blanco. Son colocaciones de frase largas. Algunas colocaciones de este tipo, en el dominio de la bolsa, se muestran a continuación:

*En la bolsa de valores americana el índice del valor comercial estaba encima de *NUMERO**
*La tasa promedio acabó la semana con una pérdida neta de *NUMERO**
*La tasa promedio Dow Jones de treinta industrias bajo de *NUMERO* a *NUMERO* puntos*

En las colocaciones anteriores, las ranuras vacías deben ser llenadas con un número (indicado por *NUMERO* en los ejemplos). Más generalmente, las plantillas de frase especifican las categorías gramaticales de las palabras que pueden llenar las ranuras vacías.

Las plantillas de frase son absolutamente representantes de un dominio dado y se repiten muy a menudo de una manera rígida en un sublenguaje dado. Son específicamente útiles para generación de texto.

2.3 Aplicaciones de colocaciones

Como se ha mencionado antes, las colocaciones son útiles en diversas aplicaciones de procesamiento de lenguaje natural. Entre las más significativas tenemos [9]:

- Redacción de texto
- Resolución de ambigüedad sintáctica o análisis sintáctico
- Desambiguación de sentidos de palabras
- Detección y corrección de malapropismos
- Traducción automática
- Reconocimiento de cohesión de texto
- Segmentación en párrafos
- Esteganografía lingüística

A continuación se presenta una breve descripción de cada una de ellas.

Redacción de texto

Una de las aplicaciones principales de las colocaciones es ayudar a cualquier autor a redactar un texto, seleccionando palabras que combinen sintáctica y semánticamente. Hay sistemas que se encargan de llevar a cabo esta tarea automáticamente y se conocen como sistemas generadores de lenguaje.

Resolución de ambigüedad sintáctica o análisis sintáctico

El proceso de la resolución de ambigüedad sintáctica en utilizar conocimiento lingüístico para elegir el árbol sintáctico correcto. Este conocimiento lingüístico se encuentra en un diccionario de colocaciones.

Una idea de los pasos a seguir para este proceso se describe a continuación:

1. Una vez que se tienen los árboles sintácticos posibles de la oración, se extraen todas las relaciones sintácticas (colocaciones) de cada uno de ellos.
2. Se buscan las colocaciones en el diccionario, sumando las frecuencias de todas ellas. Si la colocación no se encuentra, entonces su frecuencia es cero.
3. Se elige el árbol sintáctico que contenga la mayor suma de frecuencias de sus colocaciones.
4. Si el diccionario de colocaciones no tuviera frecuencias entonces se consideraría el valor cero si no existe la colocación y 1 si existe.

Desambiguación de sentidos de palabras

Tomada fuera de contexto, una colocación puede tener diferentes significados, mientras que una colocación adicional puede desambiguar el sentido inmediatamente. Ejemplo, "banco" es: "dinero" si en la base de datos de colocaciones se encuentra "cuenta de banco", es "transfusión" si se encuentra "banco de sangre", es "mueble" si tenemos "sentarse en banco".

5.5.4 Detección y corrección de malapropismos

Malapropismo es un error semántico de reemplazar una palabra real por otra, similar a la deseada en sonido y función sintáctica pero distinta en significado. Ejemplo, centro histérico (queriendo decir centro histórico). Para detectar el malapropismo se propone en [19] basarse en anomalías semánticas en textos originados de los que el malapropismo usualmente destruye el contexto de colocaciones, es decir, la frase dada es sintácticamente correcta pero su colocación no.

La ausencia de tal combinación en la base de datos de colocaciones puede significar un malapropismo en el texto revisado. Para corregir el error es necesario buscar entre las palabras reales las similares a la errónea. Si un candidato restaura la colocación con las otras palabras del contexto, ésta podría ser mostrada al usuario para considerarse.

Traducción automática

Supongamos que tenemos una base de datos de colocaciones en español con una interfaz de opción de traducción. El usuario puede introducir, como consulta, una colocación correcta en un lenguaje diferente al español. Si existen en la base de datos de colocaciones, para cada colocación solicitada, una lista de sus equivalentes en español se mostrará al usuario [21]. Note que en dirección contraria una traducción correcta es generalmente irrealizable.

Reconocimiento de cohesión de texto

El texto "María comió rápidamente, las donas estaban sabrosas", parece consistente para nosotros debido a que donas es comida, esto hace claro que María las comió. Una aplicación puede emular el reconocimiento de cohesión de texto si encuentra la colocación "comer donas" en la base de datos de colocaciones.

Segmentación en párrafos

En [1] se propone un método para segmentación automática de textos en párrafos. La cohesión en la palabra actual es medida por el número de palabras que están dentro de las ligas puramente semánticas y las ligas intracolocación. Una separación de párrafo es colocada cerca mínimo local de profundad definido para la medida de cohesión.

Esteganografía lingüística

Bits de información secreta pueden ser escondidos en un texto que parece inofensivo, seleccionando sinónimos específicos de palabras en un orden previamente acordado, estableciendo la selección del primer sinónimo posible de una palabra por 0, el segundo por 1, etc. Para mantener la cohesión y naturalidad del texto, el sistema elige sólo sinónimos que forman colocaciones de palabras compatibles.

3 Formalismos sintácticos de la lingüística computacional

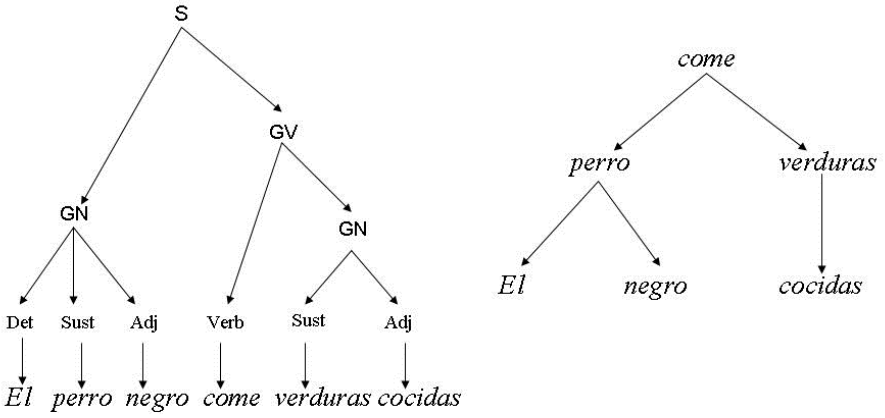
La base para la extracción de colocaciones que se presenta en este artículo es un corpus etiquetado con estructuras sintácticas. Existen dos principales formalismos para representar la estructura sintáctica de una oración: el formalismo de constituyentes (o estructura de frase) cuyo principal representante es la teoría desarrollada por Chomsky en sus diversas variantes; y la tradición estructuralista europea (dependencias) que proviene de Tesniére, con el ejemplo más representativo, la teoría Sentido ⇔ Texto de I. A. Mel'čuk.

Siguiendo el paradigma de Chomsky, se han desarrollado muchos formalismos para la descripción y el análisis sintácticos. El concepto básico de la gramática generativa es simplemente un sistema de reglas que define de una manera formal y precisa un conjunto de secuencias (cadenas a partir de un vocabulario de palabras) que representan las oraciones bien formadas de un lenguaje específico. Las gramáticas bien conocidas en otras ramas de la ciencia de la computación, las expresiones regulares y las gramáticas libres de contexto, son gramáticas generativas también.

Chomsky y sus seguidores desarrollaron y formalizaron una teoría gramatical basada en la noción de generación [22]. El trabajo que se realiza en la gramática generativa descansa en la suposición acerca de la estructura de la oración que está organizada jerárquicamente en frases (y por consiguiente en estructura de frase). Un ejemplo de la segmentación y clasificación que se realiza en este enfoque se presenta en la figura 1A en el árbol de constituyentes para la frase "El perro negro come verduras cocidas".

Un árbol de constituyentes revela la estructura de una expresión en términos de agrupamientos (bloques) de palabras, que consisten de bloques más pequeños, los cuales consisten de bloques aún más pequeños, etc. En un árbol de constituyentes, la mayoría de los nodos representan agrupamientos sintácticos o frases, y no corresponden a las formas de las palabras reales de la oración bajo análisis. Símbolos como S (oración), GN (grupo nominal), GV (grupo verbal), Sust (sustantivo), GP (grupo preposicional), etc. aparecen en los árboles de constituyentes como etiquetas en los nodos, y se supone que estas únicas etiquetas completamente determinan las funciones sintácticas de los nodos correspondientes.

En el enfoque de constituyentes(o estructura de frase), la categorización (la membresía de clase sintáctica) de las unidades sintácticas se especifica como una parte integral de la representación sintáctica, pero no se declaran explícitamente las relaciones entre unidades.



A) Árbol de constituyentes **B) Árbol de dependencias**

Fig. 1. Estructuras sintácticas

Las gramáticas de dependencias se basan en la idea de que la sintaxis es casi totalmente una materia de capacidades de combinación, y en el cumplimiento de los requerimientos de las palabras solas. En el trabajo más influyente en este enfoque, se presenta en [23], el modelo para describir estos fenómenos es semejante a la formación de moléculas, a partir de átomos, en la química. Como átomos, las palabras tienen valencias; están aptas para combinarse con un cierto número y clase de otras palabras, formando piezas más grandes de material lingüístico.

Las valencias de una palabra se rellenan con otras palabras, las cuales realizan dos tipos de funcionamiento: principales (denominadas actuantes) y auxiliares (denominados circunstanciales o modificadores). Las descripciones de valencias de palabras son el dispositivo principal para describir estructuras sintácticas en las gramáticas de dependencias.

La gramática de dependencias supone que hay comúnmente una asimetría entre las palabras de una frase: una palabra es la rectora, algunas otras son sus dependientes. Cada palabra tiene su rectora, excepto la raíz, pero no todas tienen dependientes. Por ejemplo, una palabra es "verduras", la modificadora es "cocidas". La palabra rectora raíz da origen a la construcción total y la determina. Las dependientes se ajustan a las demandas sobre la construcción, impuestas por la rectora. La diferencia entre rectoras y dependientes se refleja por la jerarquía de nodos en el árbol de dependencias.

Las gramáticas de dependencia, como las gramáticas de constituyentes, emplean árboles a fin de describir la estructura de una frase u oración completa. Mientras la gramática de constituyentes asocia los nodos en el árbol con constituyentes mayores o menores y usa los arcos para representar la relación entre una parte y la totalidad, todos los nodos en un árbol de dependencias representan palabras elementales y los arcos denotan las relaciones directas sintagmáticas entre esos elementos (Figura 1B).

Las teorías de constituyentes y las gramáticas de dependencias se han desarrollado en paralelo. Ambas han marcado la forma en la que se concibe la sintaxis en el procesamiento lingüístico de textos. A lo largo de casi cuarenta años, muchos formalismos se han desarrollado dentro de ambos enfoques de una manera muy diferente. A continuación se presenta una un panorama del desarrollo de ambos formalismos.

3.1 Formalismos de constituyentes

Chomsky [24] presentó una versión inicial de la Gramática Generativa Transformacional (GGT), gramática en la cual, la sintaxis se conoce como sintaxis generativa. Una de las características del análisis presentado ahí y en subsecuentes trabajos transformacionales es la inclusión de postulados explícitos formales en las reglas de producción, cuyo único propósito era genera r todas las oraciones gramaticales del lenguaje bajo estudio, es decir, del inglés.

La gramática transformacional inicial influyó, a las teorías posteriores, en el énfasis en la formulación precisa de las hipótesis, característica primordial en el enfoque de constituyentes. Ejemplos de las reglas de producción que se emplean para esa formulación precisa son las siguientes, con las cuales se construyó el árbol de la figura 1A.

O	→	GN GV		ADJ	→	negro	|	cocidas
GN	→	ART SUST ADJ		SUST	→	perro	|	verduras
GN	→	ADJ SUST		V	→	come		
GV	→	V GN		ART	→	el		

La flecha significa que se reescribe como, es decir, el elemento de la izquierda se puede sustituir con el agrupamiento completo de la derecha. Por ejemplo, una oración (O) se puede reescribir como un grupo nominal (GN) seguido de un grupo verbal (GV). Un GN puede reescribirse como un artículo (ART) seguido de un sustantivo (SUST) y un adjetivo (ADJ). Un grupo verbal puede sustituirse con un verbo (V) seguido de un grupo nominal. Todos los elementos que no han sido sustituidos por palabras específicas se denominan no-terminales (GV, O, etc.), los elementos del lenguaje específico se denominan terminales (come, perro, etc.).

Este tipo de reglas corresponde a una gramática independiente del contexto. Esto se debe a que los elementos izquierdos de las reglas solamente contienen un elemento no terminal y por lo tanto no se establece el contexto en el que deben aparecer. Este tipo de gramáticas es el segundo tipo de gramáticas menos restrictivas en la clasificación de Chomsky, que pueden analizarse con un autómata de pila, y para las cuales existen algoritmos de análisis eficientes [25].

Chomsky dio varios argumentos para mostrar que se requería algo más que las solas reglas de estructura de frase para dar una descripción razonable del inglés, y por extensión, de cualquier lenguaje natural, por lo que se requerían las transformaciones, es decir, reglas de tipos más poderosos.

La GGT define oraciones gramaticales de una manera indirecta. Las estructuras aquí denominadas subyacentes o base se generan mediante un sistema de reglas de estructura de frase y después se aplican sucesivamente las reglas transformacionales

para mapear esas estructuras de frase a otras estructuras de frase. Esta sucesión se llama derivación transformacional e involucra una secuencia de estructuras de frase, de una estructura base a una estructura de frase denominada estructura superficial, cuya cadena de palabras corresponde a una oración del lenguaje. Desde este punto de vista, las oraciones del lenguaje son aquellas que pueden derivarse de esta manera.

Una propuesta clave en las gramáticas transformacionales, en todas sus versiones, es que una gramática empíricamente adecuada requiere que las oraciones estén asociadas no con una sola estructura de árbol sino con una secuencia de árboles, cada una relacionada a la siguiente por una transformación. Las transformaciones se aplican de acuerdo a reglas particulares en forma ordenada; en algunos casos las transformaciones son obligatorias.

Otro punto muy importante de la GGT fue el tratamiento del sistema de verbos auxiliares del inglés, el análisis más importante en esta teoría. La GGT inicial se transformó con base a los cambios propuestos en los trabajos de [26] y de [22]. La teoría resultante fue la Teoría Estándar (*Standard Theory*, ST). Entre esos cambios, la ST introdujo el uso de reglas recursivas de estructura de frase para eliminar las transformaciones que combinaban múltiples árboles en uno solo, y la inclusión de características sintácticas, para considerar la subcategorización. Otra aportación fue la adición de una componente semántica interpretativa a la teoría de la gramática transformacional.

Chomsky abandonó algunas ideas de la ST y propuso la Teoría Estándar Ampliada (*The Extended Standard Theory*, EST), una teoría muy reducida en transformaciones, en su lugar se mejoraron otras componentes de la teoría para mantener la capacidad descriptiva. Además de nuevos tipos de reglas semánticas, introdujeron la esquematización de reglas de estructura de frase, y una concepción mejorada del diccionario, incluyendo reglas léxicas. Estas modificaciones se han trasladado a muchos trabajos contemporáneos.

La EST presentó dos modificaciones esenciales:

- El modelo de interpretación semántica debe considerar el conjunto de árboles engendrados por las transformaciones a partir de la estructura profunda.
- El modelo incluye una etapa de inserción léxica antes de la aplicación de las transformaciones. Así que sólo existen dos tipos de reglas: las gramaticales y las de inserción léxica.

Las teorías siguientes a partir de la EST buscaron sobre todo resolver las cuestiones metodológicas debidas a la sobrecapacidad del modelo. [27] y [28] demostraron que el modelo transformacional era equivalente a una gramática sin restricciones.

De hecho, después de varios años de trabajo, esta ba claro que las reglas transformacionales eran muy poderosas y se permitían para toda clase de operaciones que realmente nunca habían si do necesarias en las gramáticas de lenguajes naturales. Por lo que el objetivo de restringir las transformaciones se volvió un tema de investigación muy importante.

Con base en esto Bresnan [29] presenta la Gramática Transformacional Realista que por primera vez proveía un tratamiento convincente de numerosos fenómenos, como la posibilidad de tener forma pasiva en términos léxicos y no en términos trans-

formacionales. Este paso de Bresnan fue se guido por otros investigadores para tratar de eliminar totalmente las transformaciones en la teoría sintáctica.

Otra circunstancia en favor de la eliminación de las transformaciones fue la introducción de la Gramática de Montague [30, 31], ya que al proveer nuevas técnicas para la caracterización de los sentidos, directamente en términos de la estructura superficial, eliminaba la motivación semántica para las transformaciones sintácticas. Con el empleo de métodos de análisis semántico como el de Montague, se podían asignar formalmente distintas estructuras superficiales a distintas pero equivalentes interpretaciones semánticas; de esta manera, se consideraba la semántica sin necesidad de las transformaciones.

Es así como a fines de la década de los setenta y principios de los ochenta surgen los formalismos generativos donde las transformaciones, si existen, tienen un papel menor. Los más notables entre estos son: *Government and Binding* (GB), *Generalized Phrase Structure Grammar* (GPSG), *Lexical-Functional Grammar* (LFG) y *Head-Driven Phrase Structure Grammar* (HPSG), que indican los caminos que han llevado al estado actual en el enfoque de constituyentes.

3.2 Formalismos de dependencias

Mel'cuk [32] explicó que un lenguaje de constituyentes describe muy bien cómo los elementos de una expresión en lenguaje natural combinan con otros elementos para formar unidades más amplias de un orden mayor, y así sucesivamente. Un lenguaje de dependencias, por el contrario, describe cómo los elementos se relacionan con otros elementos, y se concentra en las relaciones entre unidades últimas sintácticas, es decir, entre palabras.

La estructura de un lenguaje también se puede describir mediante árboles de dependencias, los cuales presentan las siguientes características:

- Muestra cuáles elementos se relacionan con otros y en qué forma.
- Revela la estructura de una expresión en términos de ligas jerárquicas entre sus elementos reales, es decir, entre palabras.
- Se indican explícitamente los roles sintácticos, mediante etiquetas especiales.
- Contiene solamente nodos terminales, no se requiere una representación abstracta de agrupamientos.

Con las dependencias se especifican fácilmente los tipos de relaciones sintácticas. Pero la membresía de clase sintáctica (categorización) de unidades de orden más alto (GN, GP, etc.) no se establece directamente dentro de la representación sintáctica misma, así que no hay símbolos no-terminales en representaciones de dependencias.

Una gramática cercana a este enfoque de dependencias es la Gramática Relacional (*Relational Grammar, RG*) [33] que adopta primitivas que son conceptualmente muy cercanas a las nociones relacionales tradicionales de sujeto, objeto directo, y objeto indirecto. Las reglas gramaticales de la RG se formularon en términos relacionales, reemplazando las formulaciones iniciales, basadas en configuraciones de árboles. Por ejemplo, la regla pasiva se establece más en términos de promover el objeto directo al sujeto, que como un re-arreglo estructural de grupos nominales.

Los ejemplos más representativos de este formalismo son: *Dependency Unification Grammar* (DUG), *Word Grammar* (WG) y *Meaning ⇔ Text Theory* (MTT)

4 Extracción del diccionario de colocaciones

En este artículo se presenta un método que obtiene un diccionario de colocaciones en español a partir de un corpus etiquetado manualmente con las estructuras sintácticas. El corpus utilizado es el corpus en español Cast3LB y la extracción del diccionario de colocaciones puede ser descrita en dos pasos:

1. Transformación del corpus de constituyentes a corpus de dependencias
2. Extracción de colocaciones
3. Agregar información estadística

4.1 El corpus en español Cast3LB

Cuenta con cien mil palabras (aproximadamente 3,500 oraciones) creado a partir de dos córporas: el corpus CLiCTALP (75,000 palabras), un corpus balanceado y anotado morfológicamente que contiene un lenguaje literario, periodístico, científico, etc.; y el corpus de la agencia de noticias española EFE (25,000 palabras) correspondiente al año 2000.

El proceso de anotación se llevó a cabo en dos pasos. En el primero, un subconjunto del corpus ha sido seleccionado y anotado dos veces por dos diferentes anotadores. Los resultados de este proceso de doble anotación se han comparado y una topología de desacuerdo en asignación de sentido ha sido establecida. Después de un proceso de análisis y discusión, un manual de anotación ha sido producido, donde los criterios principales a seguir en caso de ambigüedad se han descrito. En el segundo paso, el resto del corpus ha sido anotado siguiendo todas las estrategias de palabras. Los items léxicos anotados son esas palabras con significado léxico, es decir, sustantivos, verbos y adjetivos [34].

4.2 Transformación del corpus de constituyentes a corpus de dependencias

Al igual que la mayoría de las herramientas y recursos existentes, el corpus Cast3LB se orienta a la representación de constituyentes. La extracción de colocaciones (relaciones sintácticas) se hace en base a estructuras sintácticas orientadas a dependencias., es por ello que para llevar a cabo la extracción automática del diccionario de colocaciones el primer paso es transformar el corpus de constituyentes a un corpus de dependencias.

Para esta transformación se llevó a cabo el proceso descrito en [35] y que, en general, se describe a continuación:

1. Extracción de las reglas gramaticales del corpus de constituyentes Cast3LB
2. Determinación de rectores o cabezas de cada regla gramatical mediante el uso de heurísticas

3. Utilizar la información de rectores o cabezas, de forma recursiva, para determinar cuáles reglas y componentes se subirán de nivel en el árbol de dependencias.

Además, como parte de este proceso, se da un tratamiento especial a pronombres y conjunciones mismo que se describe en [36].

4.3 Extracción de colocaciones

Una vez que tenemos el corpus de dependencias aplicamos los siguientes pasos para extraer las colocaciones:

1. Recorremos el árbol de dependencias en profundidad de izquierda a derecha, comenzando de la raíz.
2. Por cada nodo hijo del nodo visitado, se extrae el nodo padre, el nodo hijo y la relación de dependencia entre ellos. Si el nodo hijo es una preposición entonces éste se considera como la relación de dependencia y el nodo hijo de la preposición se considera el nodo hijo de la colocación.

No se consideran las colocaciones donde existen determinantes (artículos) debido a que, como mencionamos anteriormente, las colocaciones son pares de palabras de contenido con su relación sintáctica.

Para ilustrar un ejemplo consideramos la siguiente oración: "Los policías velarán por la seguridad de los líderes" (el árbol de dependencias extraído de esta oración se muestra en la figura 2).

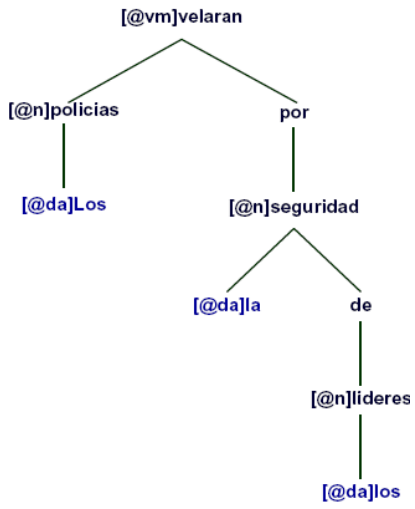


Fig. 2. Árbol sintáctico de dependencias para la oración "Los policías velarán por la seguridad de los líderes"

Recorriendo el árbol (de la figura 2), visitamos el primer nodo que sería la raíz y encontramos que la primer colocación a extraer es *velarán SUST policías*, donde *vela-*

rán es el nodo padre, *policías* es el nodo hijo y *SUST* es la relación de dependencia entre ellos. Las colocaciones extraídas automáticamente de la oración "Los policías velarán por la seguridad de los líderes" son las siguientes y se muestran encerradas en óvalos en la figura 3:

seguridad de líder

velar SUST policía

velar por seguridad

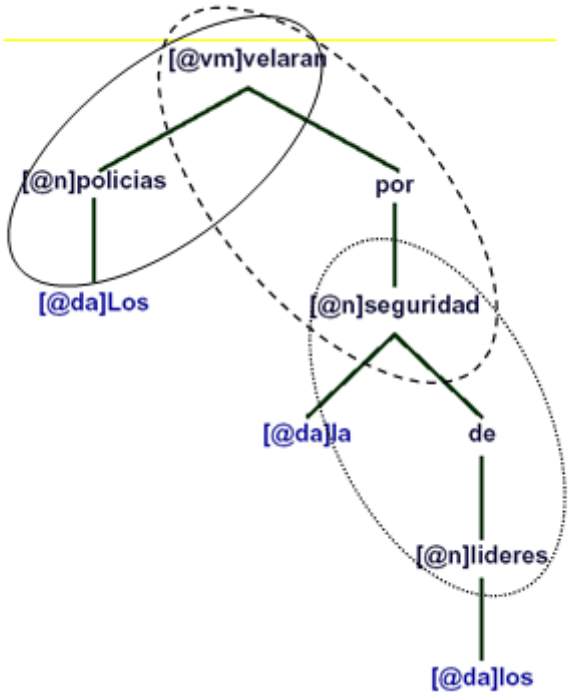


Fig. 3. Colocaciones extraídas del árbol de dependencias de la oración "Los policías velarán por la seguridad de los líderes".

4.4 Agregar información estadística

Por último, para las frecuencias de las colocaciones del diccionario se llevaron a cabo los siguientes pasos:

1. Se ordenaron las colocaciones obtenidas
2. Se cuentan las frecuencias de las colocaciones
3. Se eliminan las colocaciones repetidas

Esta información fue agregada ya que se considera importante para algunas aplicaciones de procesamiento de lenguaje natural, específicamente, para resolver la ambigüedad sintáctica.

5 Resultados

El diccionario de colocaciones en español extraído automáticamente del corpus Cast3LB consta de 40,121 colocaciones únicas. Cada una de esas colocaciones está formada por la palabra rectora (o nodo padre), la palabra dependiente (nodo hijo) y la relación sintáctica entre ambas, así como al frecuencia de dicha colocación.

En el diccionario se encuentran tres tipos de colocaciones: 13,048 oraciones nominales rígidas, 423 plantillas de frase y 26,650 relaciones predicativas.

Con respecto a las colocaciones de tipo relaciones predicativas, en el diccionario se clasifican de acuerdo a los siguientes tipos de relación sintáctica entre palabras: sustantivo, adjetivo, verbo, adverbio, pronombre, coordinante, negación y preposición.

La tabla siguiente muestra el número de colocaciones que contiene el diccionario con respecto al tipo de colocación y la relación sintáctica correspondiente, así como un ejemplo para cada caso.

Tabla 3. Colocaciones extraídas automáticamente de acuerdo al tipo de colococación

Tipo de colocación	Relación sintáctica de la colocación	Número de colocaciones obtenidas	Ejemplo
Relaciones predicativas	Sustantivo	8,821	1 aceptar SUST suerte
	Adjetivo	5,802	1 águila ADJ real
	Verbo	5,370	1 acuerdo VERB establecer
	Adverbio	2,692	1 acabar ADV bien
	Pronombre	2,103	1 acabar PRON ese
	Coordinante	1,364	1 prestar COORD si
	Negación	293	1 aprovechar NEG no
	Preposición	205	1 arrinconar PREP contra
Oraciones nominales rígidas	----	13,048	1 ansia de revolución
Plantillas de frase	----	423	1 alcanzar CIF *número* Antes_de FECH *fecha*

Con relación a la información estadística del diccionario, 36,501 colocaciones tienen frecuencia de 1, es decir, sólo aparecen una vez en el corpus; 3,524 colocaciones tienen frecuencia de 2 a 10; 94 tienen frecuencia de 11 a 100, mientras que sólo dos colocaciones contienen una frecuencia mayor a 100.

5.1 Evaluación

Para evaluar las colocaciones obtenidas automáticamente, se seleccionaron dos muestras. Cada una consiste de 17 oraciones seleccionadas aleatoriamente de todas las oraciones del corpus Cast3LB.

El sistema extrae automáticamente las colocaciones de cada oración en ambas muestras. Para poder compararlas, un experto extrajo manualmente las colocaciones de las oraciones de cada muestra.

La tabla 4 muestra los resultados de las colocaciones extraídas manualmente, automáticamente, colocaciones que coinciden, la precisión y el *recall* por cada oración de la primera muestra seleccionada.

Tabla 4. Resultados obtenidos para la primer muestra

Oración	Colocaciones extraídas manualmente	Colocaciones extraídas automáticamente	Colocaciones que coinciden	Precisión	*Recall*
1	15	15	13	86.7	86.7
2	28	30	21	70.0	75.0
3	9	9	8	88.9	88.9
4	12	12	11	91.7	91.7
5	11	11	10	90.9	90.9
6	6	6	6	100	100
7	9	9	5	55.6	55.6
8	15	15	15	100	100
9	11	11	11	100	100
10	17	17	16	94.1	94.1
11	9	6	6	100	66.7
12	13	10	10	100	76.9
13	7	7	7	100	100
14	14	13	12	92.3	85.7
15	12	12	12	100	100
16	15	15	12	80.0	80.0
17	8	8	8	100	100
			Promedio:	**91.2**	**87.8**

La tabla 5 contiene los mismos resultados pero de la segunda muestra seleccionada. Por precisión se entiende el porcentaje de las colocaciones extraídas automáticamente que también fueron extraídas manualmente, mientras que por *recall* se entiende el porcentaje de las colocaciones extraídas manualmente que también fueron extraídas automáticamente.

El promedio de la precisión de la unión de las dos muestras es de 89.7, con una desviación estándar de 10.2, y el promedio de *recall* es de 88.7 con una desviación estándar de 11.3. La desviación estándar entre los dos promedios de las dos muestras es 0.5% para la precisión y 0.9% para el *recall*. Entonces, para otras muestras extraí-

das del mismo corpus, el porcentaje promedio de colocaciones correctas será similar a los valores obtenidos para las muestras seleccionadas.

El diccionario extraído del corpus Cast3LB cuenta con 40,121 colocaciones. Extrapolando los resultados, inferimos que 89.7% ± 0.5% de ellas son correctas (89.7% ± 10.2% en cada oración específica), y que el diccionario extraído contiene 88.7% ± 0.9% de las colocaciones contenidas realmente en el corpus (88.7% ± 11.3% de cada oración específica).

Tabla 5. Resultados obtenidos para la segunda muestra

Oración	Colocaciones extraídas manualmente	Colocaciones extraídas automáticamente	Colocaciones que coinciden	Precisión	*Recall*
1	3	3	3	100	100
2	18	20	18	90.0	100
3	8	8	6	75.0	75
4	10	10	10	100	100
5	9	9	7	77.8	77.8
6	8	8	8	100	100
7	5	5	5	100	100
8	13	13	11	84.6	84.6
9	16	17	15	88.2	93.8
10	18	18	18	100	100
11	12	12	11	91.7	91.7
12	12	12	11	91.7	91.7
13	11	11	9	81.8	81.8
14	8	10	6	60.0	75.0
15	8	8	7	87.5	87.5
16	13	13	10	76.9	76.9
17	33	31	29	93.5	87.9
			Promedio:	**88.2**	**89.6**

6 Conclusiones y trabajo futuro

El proceso de etiquetar corpus con estructuras sintácticas es una tarea que se sigue llevando acabo. Estos recursos son la base necesaria para extraer colocaciones automáticamente y no enfocar esfuerzos en la generación manual de diccionarios de colocaciones.

En este artículo se llevó a cabo la extracción automática de un diccionario de colocaciones en español basado el corpus etiquetado en español Cast3LB. El diccionario obtenido contiene más de 40,000 colocaciones con más del 89% de precisión.

Como trabajo futuro se propone mejorar el método de extracción de colocaciones, aquí presentado, para obtener mejores resultados, un ejemplo sería el uso del "que" como preposición en algunos grupos verbales (ejemplo: "tienen que tener", donde "que" sería la relación sintáctica entre los dos verbos).

Referencias

1. Bolshakov, I.A., Gelbukh, A.: A very large database of collocations and semantic links. Lecture Notes in Computer Science, N 1959, Springer, pp. 103–114 (2001).
2. Bolshakov, I.A., Gelbukh, A.: Enseñando idiomas extranjeros con una base de colocaciones. In: Caridad Anías Calderón, En: La telemática y su aplicación en la educación a distancia y en la informatización de la sociedad. Editorial "Félix Varela", Cuba, Tomo II, p. 632–638 (2002).
3. Nakhimovsky, A. D., Leed, R. L.: Lexical functions and language learning. Slavic and East European Journal (1979).
4. Allerton, D. J.: Three or four levels of co-occurrence relations. Lingua, 63, pp. 17-40 (1984).
5. Cruse, D. A.: Lexical Semantics. Cambridge University Press. Cambridge, United Kingdom (1986).
6. Mel'cuk, I. A.: Meaning-Text models: A recent trend in Soviet linguistics. In: Annual Review of Anthropology 10, pp. 27-62 (1981).
7. Gelbukh, A., Kolesnikova, O: Semantic Analysis of Verbal Collocations with Lexical Functions. Springer, XI + 146 pp. (2013).
8. Benson, M.: Collocations and general-purpose dictionaries. In: International Journal of Lexicography, 3(1), pp. 23-35 (1990).
9. Bolshakov, I. A.: Getting One's First Million… Collocations. In: CICLing 2004, LNCS 2945 pp. 229-242 (2004).
10. Dellenbaugh, D., Dellenbaugh, B.: Small Boat Sailing, a Complete Guide. Sports Illustrated Winner's Circle Books (1990)
11. Smadja, F. A.: Retrieving Collocations from Text: Xtract. In: Computational Linguistics 19.1: pp. 143-176 (1993)
12. Cowie, A. P.: The treatment of collocations and idioms in learner's dictionaries. In: Applied Linguistics, 2(3), pp. 223-235 (1981)
13. Benson, M.: The collocational dictionary and the advanced learner. In: Learner's Dictionaries: State of the Art, edited by M. Tickoo, 84-93. SEAMEO, (1989).
14. Sidorov, G.: Métodos de análisis de la combinabilidad de palabras en ruso (en ruso). In: Taal en cultur, Maastricht, Holanda y Moscú, Rusia, pp. 294-302 (1999).
15. Benson, M., Benson, E., Ilson, R.: The BBI combinatory dictionary of English: a guide to word combinations. John Benjamins Publishing Company, Philadelphia (1986).
16. Halliday, M. A. K.: Lexis as a linguistic level. In: In Memory of J. R. Firth, edited by C. E. Bazell, J. C. Catford, M. A. K. Halliday, and R. H. Robins, Longmans Linguistics Library, pp. 148-162, (1966).
17. Choueka, Y.: Looking for needles in a haystack. In: Proceedings, RIAO Conference on User-Oriented Context Based Text and Image Handling, pp. 609-623. Cambridge, MA (1988).
18. Amsler, B.: Research towards the development of a lexical knowledge base for natural language processing. In: Proceedings, 1989 SIGIR Conference. Cambridge, MA (1989).
19. Bolshakov, I. A., Gelbukh, A.: On detection of Malapropisms by Multistage Collocation Testing. In: NLDB-2003, 8th Int. Conf. on Application on Natural Lengauge to Information Systems. Bonner Köllen Verlag, pp. 28-41 (2003).
20. Bolshakov, I. A., Gelbukh, A.: Text segmentation to Paragraphs based on Local Text Cohesion. In: V. Matousek et al. (Eds). Text Speech and Dialogue. Proc. 4th Intern. Conf. TSD-2001. Lecture Notes in Artificial Intelligence, 2166, Springer, pp. 158-166 (2001).

21. Bolshakov, I.A., Gelbukh, A.: A Large Database of Collocations and Semantic References: Interlingual Applications. International Journal of Translation, Vol. 13, No.1–2, pp. 167–187 (2001).
22. Chomsky, N.: Aspects of the Theory of Syntax. MIT Press, Cambridge, MA (1965).
23. Tesniere, L.: Elements de syntaxe structural. Paris: Klincksiek. (German: Tesniere, L. (1980): Grundzüge der strukturalen Syntax. Stuttgart: Klett-Cotta.) (1959).
24. Chomsky, N.: Syntactic Structures. The Hague: Mouton & Co (1957)
25. Aho, A. V., Sethi, R., Ullman, J. D.: Compilers. Principles, Techniques and Tools. Addison Wesley Publishing Company, (1986).
26. Katz, J. J., Postal, P. M.: An Integrated Theory of Linguistic Descriptions. Cambridge, Mass. MIT Press, (1964).
27. Salomaa, A.: The generative power of transformational grammars of Ginsburg and Partee. Information and Control, 18, pp. 227-232 (1971).
28. Peters, P. S., Ritchie, R. W.: On the generative power of transformational grammars. Information Science, 6, pp. 49 - 83 (1973).
29. Bresnan, J. A.: Realistic transformational Grammar. In M. Halle, J. Bresnan and G. A. Miller (eds.), Linguistic Theory and Psychological Reality. Cambridge, Mass. MIT Press (1978).
30. Montague, R.: Universal Grammar. Theoria 36, pp. 373- 398 (1970).
31. Montague, R.: Universal Grammar. En: Richard Thomason (eds.), Formal Philosophy. New Haven: Yale University Press (1974).
32. Mel'cuk, I. A.: Dependency Syntax. In: P. T. Roberge (ed.) Studies in Dependency Syntax. Ann Arbor: Karoma pp. 23-90 (1979)
33. Perlmutter, D. N.: Studies in Relational Grammar I. Chicago: University of Chicago Press (1983).
34. Navarro, B., Civit, M., Martí, M.A., Marcos, R., Fernández, B.: Syntactic, Semantic and Programatic Annotation in Cast3LB. In: Shallow Processing of Large Corpora (SProLaC), a Workshop on Corpus linguistic, Lancaster, UK (2003).
35. Gelbukh, A., Torres, S., Calvo, H.: Transforming a constituency Treebank into a dependency Treebank. Procesamiento del lenguaje natural, vol. 35, pp, 145–152, España (2005).
36. Gelbukh, A., Torres, S.: Tratamiento de ciertos pronombres y conjunciones en la transformación de un corpus de constituyentes a un corpus de dependencias. In: Avances en la Ciencia de la Computación.. (Eds.) Arturo Hernández Aguirre, Jose Luis Zechinelli Martini, pp. 293-298, México (2006).

Evaluación de las herramientas comerciales y métodos del estado del arte para la generación de resúmenes extractivos individuales

Griselda Areli Matias Mendoza[1], Yulia Ledeneva[1],
René Arnulfo García-Hernández[1], Grigori Sidorov[2]

[1] Universidad Autónoma del Estado de México
Unidad Académica Profesional Tianguistenco
Instituto Literario #100, Col. Centro, Toluca, 50000, Estado de México
griselda.matias.mendoza@gmail.com, yledeneva@yahoo.com,
renearnulfo@hotmail.com
[2] Laboratorio de Lenguaje Natural y Procesamiento de Texto,
Centro de Investigación en Computación, Instituto Politécnico Nacional, DF 07738,
México
sidorov@cic.ipn.mx

Resumen. Hoy en día se tiene conocimiento del avance significativo de las herramientas y métodos del estado de arte para la generación de resúmenes extractivos individuales (GREI). Para evaluar la calidad de las herramientas y los métodos de GREI es posible comparar los resúmenes generados automáticamente con los resúmenes generados por humanos. Utilizando el mismo corpus (DUC-2002) y el mismo método de evaluación independiente (ROUGE), en este trabajo se evaluaron seis herramientas comerciales y siete métodos del estado del arte que se desarrollaron para GREI. Con respecto a las herramientas comerciales de GREI, interesantemente se puede observar el avance significativo de las herramientas en línea con respecto a las herramientas instalables.

Palabras Clave: Resumen automático, Svhoong Summarizer, Pertinence Summarizer, Tool4noobs Summarizer, Open Text Summarizer.

1 Introducción

Actualmente la información en formato digital se ha ido incrementando de forma exponencial, se estima que crece a un ritmo de 57% anual [1]. El incremento acelerado de la información dificulta la obtención de documentos sobre temas específicos. Por lo que sería conveniente que los documentos recuperados de una consulta en Internet fueran relevantes y no redundantes [2]. Sin embargo, normalmente no es así, por lo que es necesario contar con herramientas de procesamiento automático de texto para generar resúmenes.

Un resumen tiene como objetivo presentar la información más importante en un texto más corto que el original; manteniendo las ideas principales [3]. Cuando se tiene como entrada un documento se puede generar un solo resumen, pero si la entrada fuera una colección de documentos se puede generar un resumen de todos ellos [4]. Según por su estrategia de condensación, los resúmenes se clasifican en extractivos y abstractivos [5].

Un resumen por extracción es aquel que está compuesto íntegramente por partes del documento original. Por el contrario, un resumen por abstracción puede contener texto que no se encuentra en el documento original. Generalmente, los humanos realizan resúmenes de tipo abstractivo, mientras que la mayoría de las herramientas comerciales y también los métodos del estado del arte generan resúmenes de forma extractiva.

Las herramientas comerciales que permiten la generación de resúmenes extractivos individuales (GREI) se pueden instalar en equipos propios o ejecutar en línea desde Internet. Algunas de las herramientas instalables son *Copernic Summarizer* [6] y Microsoft Office Word. Entre las herramientas en línea se encuentran *Svhoong Summarizer* [7], *Pertinence Summarizer* [8], *Tool4noobs Summarizer* [9] y *Open Text Summarizer* [10].

Además de las herramientas comerciales se tienen métodos propuestos de investigación sobre GREI, entre los cuales se encuentran, Secuencias Frecuentes Maximales (SFMs) [4], agrupamiento con Secuencias Frecuentes Maximales [11], SFM (1best first)[12], *TextRank* [13] y por último los basados en algoritmos genéticos [5].

También se tienen las heurísticas *Baseline* [14] y *Baseline:aleatorio* [15] las cuales se utilizan para medir el avance que presentan los métodos del estado del arte. Sin embargo, poco se sabe de la calidad que tienen las herramientas comerciales en comparación con los métodos propuestos en el estado del arte de GREI. En especial, las herramientas en línea son de interés para la investigación de GREI porque su diseño debe ser eficiente tanto en espacio como en tiempo ya que pueden ser ejecutados simultáneamente por varios usuarios. En cambio las herramientas instalables tienen métodos exhaustivos que pueden disponer de todos los recursos de la computadora para su trabajo.

Por ello, el interés en este trabajo de conocer la calidad de las herramientas en línea de GREI en comparación con las herramientas instalables y con los métodos del estado del arte.

Es posible conocer cuál es la calidad de las herramientas en comparación con los métodos del estado del arte, si se utiliza la misma colección de documentos y se evalúan sus resúmenes generados en comparación con los humanos. En este trabajo se utilizó la colección de documentos estándar DUC-2002, la cual está compuesta por 567 noticias en inglés de diferentes temas y la herramienta de evaluación ROUGE, que mide el parecido del resumen automático en comparación con el generado por el humano.

El artículo está organizado de la siguiente forma. En la sección 2, se describe el trabajo relacionado a este artículo. En la sección 3 se describen las

herramientas en línea que se utilizan en este trabajo para la comparación de GREI.

En la sección 4 se describe la colección de documentos y la herramienta de evaluación que se utilizará. En la sección 5 se muestran los resultados de los experimentos realizados con las herramientas comerciales en comparación con los métodos del estado del arte. Finalmente en la sección-6 se muestran las conclusiones del trabajo realizado.

2 Trabajo Relacionado

En el área del Procesamiento del Lenguaje Natural ha habido un gran avance en la creación de nuevos métodos de GREI. Sin embargo, poco se sabe sobre la calidad de las herramientas comerciales para la comparación de GREI. Se tiene conocimiento de trabajos como el de [16] y el de [17] donde se comparan herramientas comerciales. En el trabajo de [16] se comparan los resúmenes de una colección de artículos científicos por medio de las herramientas comerciales: *Microsoft Word*, *SweSum, Shvoong* y *Online Brevity Document Summarizer*. La forma de evaluación que se utiliza en (Vlainic, el al, 2013), consiste en que un grupo de experto son lo que comparan los resúmenes generados por las herramientas con el resumen del artículo. Cabe mencionar que los trabajos generados por los métodos del estado del arte utilizan en su mayoría colecciones de documentos estándar. Por lo que en este trabajo solo se tomará como referencia el de García [17], donde utiliza la colección de documentos DUC-2002 y la herramienta de evaluación ROUGE.

Las herramientas comerciales que se utilizaron en el trabajo de García [17] son: *Copernic Summarizer* y *Microsoft Office Word* en sus versiones 2003[1] y 2007[22]. *Copernic Summarizer* es un software diseñado exclusivamente para la tarea de GREI, el cual trabaja con cuatro lenguajes (inglés, alemán, francés y español) [18]. *Microsoft Office Word* es una suite ofimática para el procesamiento y edición de texto que incluye la opción de GREI.

Entre los métodos del arte que se compararon en el trabajo de García [17] están los métodos estadísticos, los cuales son: Secuencias Frecuentes Maximales (SFMs) [4], agrupamiento con Secuencias Frecuentes Maximales [11], SFM (1best first) [12] y ponderación basada en grafos *TextRank* [13] . También se comparó con los métodos heurísticos llamados *Baseline* y *Baseline:aleatorio*; los cuales sirven de referencia para medir el avance que presentan los métodos del estado del arte. *Baseline* es una heurística que consiste en tomar las primeras *n* líneas u oraciones del texto para generar el

[1] Microsoft ® Office Word 2003. SP3 Parte de Microsoft Office Profesional Edition 2003 Copyright © 1983-2003 Microsoft Corporation.

[2] Microsoft ® Office Word 2007. Parte de Microsoft Office Professional 2007 © 2006 Microsoft Corporation.

resumen [14]. *Baseline:aleatorio* es una heurística cuyo funcionamiento consiste en tomar algunas oraciones del texto al azar [15]. Por lo que cualquier método que se comporte como *Baseline:aleatorio* no tendría razón de ser.

En el gráfico 1 se pueden observar la comparación de las herramientas instalables y los métodos del estado del arte presentados en [17].

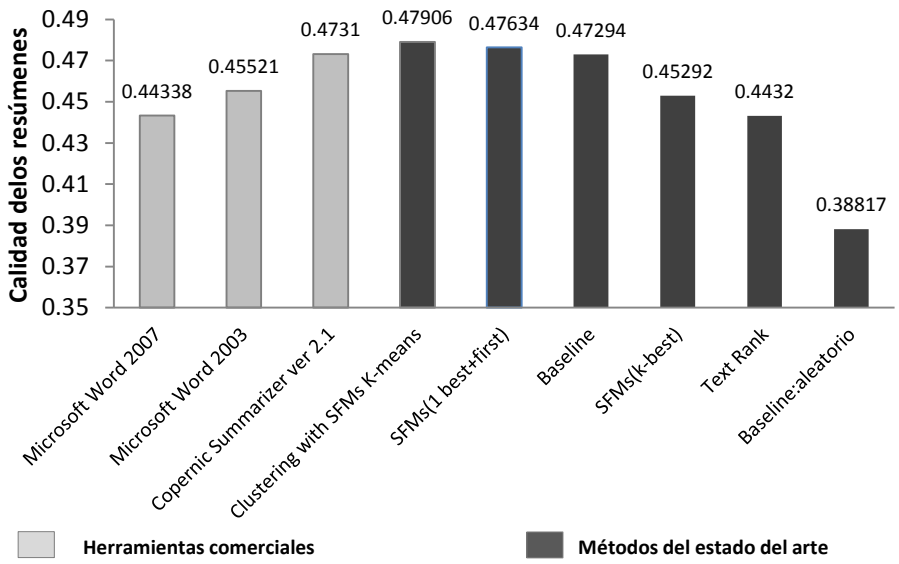


Gráfico 1. Resultados obtenidos con ROUGE 1.5.5 para la colección de resúmenes obtenidos por las herramientas comerciales y los métodos del estado del arte [17].

Tomando como base la heurística *baseline:aleatorio*, se puede observar en el gráfico 1 que tanto las herramientas comerciales instalables y los métodos del estado del arte superan esta heurística. Sin embargo, tomando como base la *Baseline*, se puede observar que la herramienta comercial instalable *Copernic Summarizer* es la única que supera ésta heurística. Los métodos del estado del arte que superan a *Baseline* son: agrupamiento con SFMs y SFMs(1 *best first*), lo que nos permite observar el avance significativo de los métodos propuestos en el estado de arte en comparación con las herramientas comerciales instalables.

Tomando como base la heurística *baseline:aleatorio*, se puede observar en el gráfico 1 que tanto las herramientas comerciales instalables y los métodos del estado del arte superan esta heurística. Sin embargo, tomando como base la *Baseline*, se puede observar que la herramienta comercial instalable *Copernic Summarizer* es la única que supera ésta heurística. Los métodos del estado del arte que superan a *Baseline* son: agrupamiento con SFMs y SFMs(1 *best first*),

lo que nos permite observar el avance significativo de los métodos propuestos en el estado de arte en comparación con las herramientas comerciales instalables.

Las herramientas comerciales se pueden clasificar en instalables y en línea, según el lugar de su ejecución. En el trabajo de [17] se compararon las herramientas comerciales instalables. Sin embargo, no se han evaluado los resúmenes generados por herramientas en línea.

En este trabajo se evalúan cuatro herramientas comerciales en línea de GREI, las cuales son: *Svhoong Summarizer* [7], *Pertinence Summarizer* [8], *Tool4noobs Summarizer* [9] y *Open Text Summarize* [10], mismas que se describen a continuación.

3 Herramientas en línea

Shvoong [7] fue fundado en 2005 por Avi Shaked y Avner Avrahami. Shvoong es una herramienta que permite generar resúmenes automáticos en 21 idiomas diferentes (checo, neerlandés, danés, inglés, finlandés, francés, alemán, griego, hebreo, húngaro, indonesio, italiano, malayo, noruego, polaco, portugués, rumano, ruso, español, sueco y turco).

A diferencia de otras herramientas *Shvoong* no devuelve el resumen como tal, sino que subraya el texto que considera más importante del documento original.

Pertinence Summarizer [8] pertenece a la gama de productos desarrollados con tecnología denominada KENiA© (basada en la extracción de conocimiento y arquitectura de notificación) desarrollada por la empresa francesa *Pertinence Mining*. *Pertinence* es una herramienta en línea que permite generar resúmenes en 12 idiomas (alemán, inglés, árabe, chino, coreano, español, francés, italiano, japonés, portugués, ruso y neerlandés) de los documentos de texto en formatos diversos (html, pdf, doc, rtf y txt).

Tools4Noobs [9] es una herramienta en línea que permite generar resúmenes desde 1 al 100 % del texto original. Para la generación de un resumen *Tools4Noobs* tiene 3 faces: extracción de las oraciones, identificación de las palabras claves del texto contando la relevancia de cada palabra e identificación de las oraciones de acuerdo a las palabras claves identificadas.

Open Text Summariser [10] es una aplicación de código abierto para resumir textos, que puede ser descargada de Internet de forma gratuita [19]. Sin embargo, también puede encontrarse la interfaz de ésta en línea [19]. OTS genera resúmenes automáticos en diferentes porcentajes y puede también generar resúmenes en 37 idiomas.

4 Experimentación

Para poder realizar la comparación entre las herramientas en línea y los métodos del estado del arte de GREI se utilizará la colección *Document Understanding Conference* (DUC, 2002) [20]. DUC-2002 fue creada por *National Institute of Standars and Technology (NITS)* para el uso de los investigadores en el área de generación de resúmenes, la cual está compuesta por 567 noticias en inglés de diversas longitudes y sobre diferentes temas. Cada noticia de DUC-2002 tiene dos resúmenes de 100 palabras creados por dos expertos humanos.

Para evaluar los resúmenes generados automáticamente por las herramientas comerciales se va a utilizar ROUGE 1.5.5 [21], otra posible manera de evaluar los resúmenes es presentada en [22]. La evaluación consiste en estimar el parecido de los resúmenes generados automáticamente con los dos resúmenes realizados por los expertos humanos.

Los resúmenes generados por las herramientas comerciales instalables así como las herramientas en línea fueron generados con un mínimo de 100 palabras, por lo que se analizó cada herramienta para satisfacer la longitud mínima del resumen automático.

Con el objetivo de saber cuál es la calidad de las herramientas en línea de GREI se compararon las siguientes herramientas *Svhoong Summarizer, Pertinence Summarizer, Tool4noobs Summarizer* y *Open Text Summarizer (OTS)*.

4.1 Evaluación de las herramientas en línea en comparación con las herramientas comerciales instalables

En este trabajo se compararon cuatro herramientas en línea y dos herramientas instalables. En el gráfico 2 se puede observar que las herramientas en línea *Shvoong* y *OTS* obtuvieron los mejores resultados que las herramientas de Microsoft Office Word. Sin embargo, las otras herramientas en línea *tools4Noons* y *Pertinence* fueron las que obtuvieron los más bajos resultados. No obstante el mejor resultado lo obtuvo la herramienta instalable *Copernic Summarizer*.

Cabe mencionar que en el trabajo realizado por García [17] se hicieron experimentos con la herramienta instalable *Microsoft Office Word* en sus versiones 2003 y 2007, pero los resúmenes que generó *Microsoft Office Word* dependieron del sistema operativo. Para completar el trabajo anterior, en este trabajo se generaron resúmenes con *Microsoft Office Word* en sus versiones 2003 y 2007 con el sistema operativo *Windows 7 Ultimate*.

Los resultados obtenidos con *Microsoft Office Word* instalado en *Windows 7 Ultimate* no superaron a los resúmenes en las versiones 2003 y 2007 en el sistema operativo *Windows Vista*. Por lo que el valor que se considerará para la comparación de la herramienta instalable *Microsoft Office Word* con los

métodos del estado del arte y las herramientas en línea, será el obtenido con la versión 2003 con el sistema operativo Windows Vista.

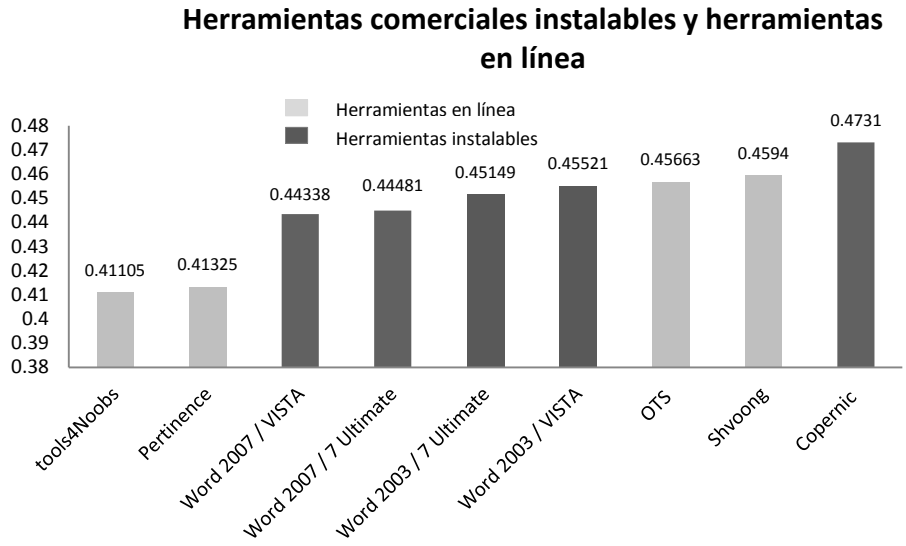


Gráfico 2. Comparación de las herramientas comerciales instalables y en línea

4.2 Evaluación de las herramientas comerciales y los métodos del estado del arte

Con el objetivo de conocer el avance que han tenido las herramientas comerciales en comparación con los métodos del estado del arte, se incluyeron los resultados anteriores junto con siete métodos del estado del arte en el gráfico 3.

En el gráfico 3 se puede observar que los resultados de las herramientas comerciales están por debajo de algunos métodos propuestos en el estado del arte. Es decir los métodos del estado de arte son de buena calidad.

Una de las heurísticas a superar por las herramientas comerciales instalables y en línea, así como por los métodos propuestos en los estados del arte es *Baseline*. Como se puede observar en el gráfico 3 solamente *Copernic Summarizer* (herramienta comercial instalable) supera esta heurística, y sólo *Clustering* con *SFM´s, SMF´s (1 best+first)* y AG-Bigramas métodos propuestos en el estado del arte superaron esta heurística.

Cabe mencionar que aunque las herramientas en línea no superan a la heurística de *Baseline*, algunas de ellas como *OTS* y *Shvoong* si están por encima de los resultados de la herramienta *Microsoft Office Word* y de los métodos del estado del arte *TextRank* y *SFM(K-best)*.

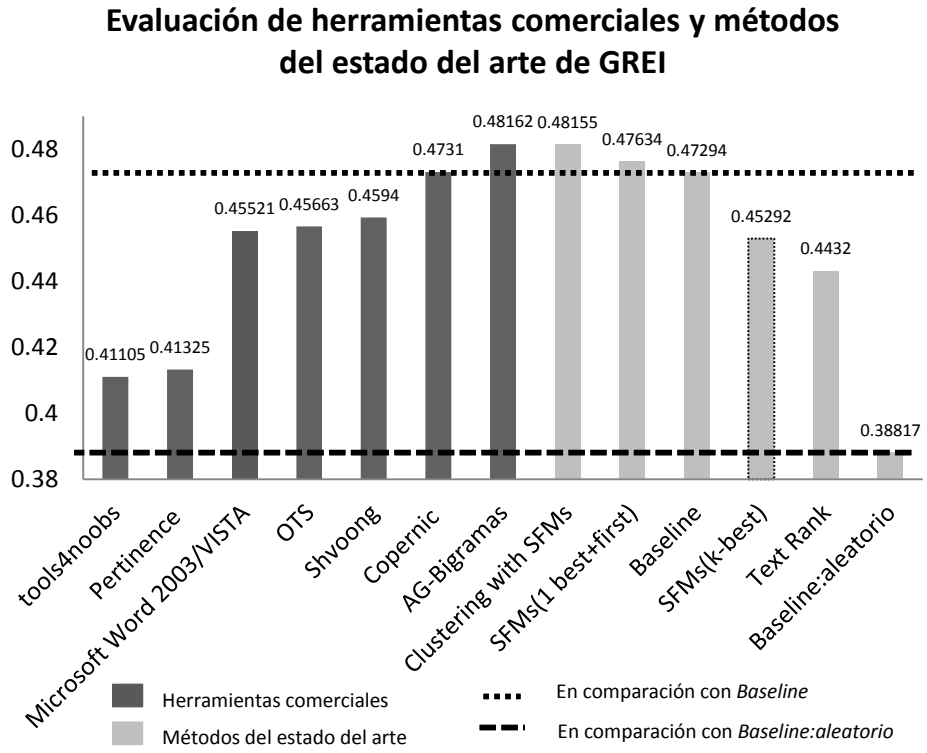


Gráfico 3. Evaluación de herramientas comerciales y métodos del estado del arte de GREI

5 Conclusiones

En este trabajo se evaluó la calidad de seis herramientas comerciales de generación de resúmenes extractivos individuales, utilizando la misma colección de noticias DUC-2002 y la misma herramienta de comparación ROUGE. La forma de trabajo anterior permitió comparar los resultados de las herramientas comerciales con siete métodos del estado del arte que utilizan la misma colección y la misma herramienta de comparación.

Por lo tanto, este trabajo contribuye con un estado del arte más amplio sobre la generación automática de resúmenes contemplando las aplicaciones y los métodos desarrollados sobre esta tarea.

En particular, se encontró que de las cuatro herramientas en línea para GREI, la mejor fue *Shvoong Summarizer*. Sin embargo, ninguna de las cuatro herramientas en línea de GREI superó a la heurística *Baseline*. Como resultado de la comparación de todas las herramientas comerciales, tanto herramientas en

línea como instalables, se observó que *Copernic Summarizer* sigue siendo la única herramienta comercial de GREI que supera a *Baseline*.

De la comparación que se muestra en el gráfico 3 se puede observar que son más los métodos del estado del arte que superan a la heurística *Baseline*. Sin embargo, aunque algunas de las herramientas comerciales no superan a *Baseline* la diferencia que existe entre estas herramientas y los métodos del estado del arte no es muy grande.

Este trabajo nos da un panorama considerando una sola colección en el lenguaje inglés, sería interesante ver si los resultados se comportarían igual considerando otros idiomas y colecciones con dominios diferentes.

Referencias

1. John F. Gantz, David Reinset, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton et al. (2007). A Forecast of Worldwide Information Growth through 2010. The Expanding Digital Universe, 4.
2. Laura Plaza Morales Uso de Grafos Semánticos en la Generación Automática de Resúmenes y Estudio de su Aplicación en Distintos Dominios: Biomedicina, Periodismo y Turismo. Madrid, España: Tesis presentada en la Facultad de Informática de la Universidad Complutense de Madrid, para obtención del grado de Doctor, 2010.
3. Saunmali, Ladda, Salim, Naomie y Binwahlan, Mohammed Salem, Genetic algorithm based sentence estraction for text summarization. 2011, International Journal of Innovative Computing, vol. 1, p. 2180-4370.
4. Yulia Ledeneva. Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization. México. D.F.: Tesis presentada en el Instituto Politécnico Nacional, para obtención del grado de Doctor, 2008.
5. Griselda Areli Matias Mendoza, Generación de Resúmenes Automáticos Usando Algoritmos Genéticos, 2013. Edo de México. Tesis de licenciatura; Unidad Académica Profesional Tianguistenco de la Universidad Autónoma del Estado de México.
6. Copernic Inc. Copernic Inc. Recuperado el 28 de 10 de 2013, de la página principal de copernic. [En línea] http://www.copernic.com/en/products/summarizer/.
7. Svhoong. Svhoong Summarizer. Recuperado el 20 de 11 de 2013, de la página principal de Svhoong. [En línea] http://es.shvoong.com/summarizer/.
8. Pertinence. Pertinence Summarizer. Recuperado el 28 de 14 de 2010, de la página principal de pertinence . [En línea] http://pertinence.net/index_en.html.
9. Tools4noobs. Tools4noobs Summarizer. Recuperado el 2 de 11 de 2013, de la página principal de tools4noobs. [En línea] http://www.tools4noobs.com/ summarize/.
10. OTS. Open Text Summarizer. Recuperado el 10 de 12 de 2013, de la página principal de OTS. [En línea] http://libots.sourceforge.net/.
11. Yulia Ledeneva, René García Hernández, Romyna Montiel Soto, Rafael Cruz Reyes, Alexander Gelbukh. EM Clustering Algorithm for Automatic Text

Summarization. 2011, Springer, Lecture Notes in Artificial Intelligence, N 7094, p. 305–315.

12. Yulia Ledeneva, Alexander Gelbukh, René García Hernández. Terms Derived from Frequent Sequences for Extractive Text Summarization. Lecture Notes in Artificial Intelligence, N 4919, 2008, Springer, pp. 593–604.
13. Rada Mihalcea, Graph-based Ranking Algorithms for Sentence Extraction. Texas; EUA: University of North Texas, 2004, Applied to Text Summarization; Department of Computer Science.
14. Esaú Villatoro Tello. Generación automática de resúmenes de múltiples documentos. Puebla: Tesis de Maestría; Instituto Nacional de Astrofísica, Óptica y Electrónica, 2006.
15. René Arnulfo García Hernández, Romyna Montiel, Yulia Ledeneva, Eréndira Rendón, Alexander Gelbukh, Rafael Cruz. Text Summarization by Sentence Extraction Using Unsupervised Learning. 2008, MICAI 2008; Lecture Notes in Artficial Intelligence, vol 5317, Springer, pp. 133–143.
16. Marta Vlainic, Nieves Mikelic Preradovic. A Comparative Study of Automatic Text Summarization System Performance, Proceedings of the 7th European Computing. 2013, pp. 222–227.
17. René Arnulfo García Hernández, Yulia Ledeneva, Griselda Areli Matias Mendoza, Ángel Hernández Domínguez, Jorge Chávez, Alexander Gelbukh, et al. Comparing Commercial Tools and State-of-the-Art Methods for Generating Text Summaries. 2009, IEEE Computer Society Press, pp. 92–96.
18. Copernic. Copernic Summarizer. Recuperado el 10 de 01 de 2014, de Copernic Summarization Technologies White Paper. [En línea] http://www.copernic.com/data/pdf/summarization-whitepaper-eng.pdf.
19. Andreas Gohr, Splitbrain.org. Recuperado el 20 de enero de 2014, de Electronic brain surgery science, 2001. [En línea] Open Text Summarizer, 2001–2013, http://www.splitbrain.org/services/ots.
20. DUC. Recuperado el 01 de 02 de 2014, de Document Understanding Conference, 2002. [En línea] 2002. http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html.
21. Chin-yew Lin, ROUGE: A package for Automatic Evaluation of Summaries. Spain: 2004, Proceedings of Workshop on Text Summarization of ACL.
22. Pinaki Bhaskar, Partha Pakray, Alexander Gelbukh, Sivaji Bandyopadhyay. Entailment-based Fully Automatic Technique for Evaluation of Summaries. Research in Computing Science, Vol. 65, 2013, pp. 11–23.

Reviewing Committee

(Comité de revisión del volumen)

Ajith Abraham
Marianna Apidianaki
Bogdan Babych
Ricardo Baeza-Yates
Kalika Bali
Sivaji Bandyopadhyay
Srinivas Bangalore
Leslie Barrett
Roberto Basili
Anja Belz
Pushpak Bhattacharyya
Igor Boguslavsky
António Branco
Nicoletta Calzolari
Nick Campbell
Michael Carl
Ken Church
Dan Cristea
Walter Daelemans
Anna Feldman
Gregory Grefenstette
Eva Hajicova
Yasunari Harada
Koiti Hasida
Iris Hendrickx
Ales Horak
Veronique Hoste
Nancy Ide
Diana Inkpen
Hitoshi Isahara
Sylvain Kahane
Alma Kharrat
Adam Kilgarriff
Philipp Koehn
Valia Kordoni
Leila Kosseim
Mathieu Lafourcade
Krister Lindén
Elena Lloret
Bente Maegaard
Bernardo Magnini
Cerstin Mahlow
Sun Maosong
Katja Markert
Diana Mccarthy
Rada Mihalcea
Jean-Luc Minel
Ruslan Mitkov
Dunja Mladenic
Marie-Francine Moens
Masaki Murata
Preslav Nakov
Vivi Nastase
Costanza Navarretta
Roberto Navigli
Vincent Ng
Kjetil Nørvåg
Constantin Orasan
Ekaterina Ovchinnikova
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Maria Pinango
Octavian Popescu
Irina Prodanof
James Pustejovsky
German Rigau
Fabio Rinaldi
Horacio Rodriguez
Paolo Rosso
Vasile Rus
Horacio Saggion
Franco Salvetti
Roser Sauri
Hinrich Schütze
Satoshi Sekine
Serge Sharoff
Grigori Sidorov
Kiril Simov
Vaclav Snasel
Thamar Solorio
Lucia Specia
Efstathios Stamatatos
Josef Steinberger
Ralf Steinberger
Vera Lúcia Strube De Lima
Mike Thelwall
George Tsatsaronis
Dan Tufis
Olga Uryupina
Karin Verspoor
Manuel Vilares Ferro
Aline Villavicencio
Piotr W. Fuglewicz
Annie Zaenen

Additional Reviewers

(Árbitros adicionales)

Rodrigo Agerri
Katsiaryna Aharodnik

Katsiaryna Aharodnik
Ahmed Ali

Tanveer Ali
Alexandre Allauzen
Maya Ando
Javier Artiles
Wilker Aziz
Vt Baisa
Alexandra Balahur
Somnath Banerjee
Liliana Barrio-Alvers
Adrián Blanco
Francis Bond
Dave Carter
Chen Chen
Jae-Woong Choe
Simon Clematide
Geert Coorman
Victor Darriba
Dipankar Das
Orphee De Clercq
Ariani Di Felippo
Maud Ehrmann
Daniel Eisinger
Ismail El Maarouf
Tilia Ellendorff
Milagros Fernández Gavilanes
Santiago Fernández Lanza
Daniel Fernández-González
Karën Fort
Koldo Gojenola
Gintare Grigonyte
Masato Hagiwara
Kazi Saidul Hasan
Eva Hasler
Stefan Hoefler
Stefan Höfler
Chris Hokamp
Adrian Iftene
Iustina Ilisei
Leonid Iomdin
Milos Jakubicek
Francisco Javier Guzman
Nattiya Kanhabua
Aharodnik Katya
Kurt Keena
Natalia Konstantinova
Vojtech Kovar
Kow Kuroda
Gorka Labaka
Shibamouli Lahiri
Egoitz Laparra
Els Lefever
Lucelene Lopes
Oier López de La Calle
John Lowe
Shamima Mithun
Tapabrata Mondal
Silvia Moraes
Mihai Alex Moruz
Koji Murakami
Sofia N. Galicia-Haro
Vasek Nemcik
Zuzana Neverilova
Anthony Nguyen
Inna Novalija
Neil O'Hare
John Osborne
Santanu Pal
Feng Pan
Thiago Pardo
Veronica Perez Rosas
Michael Piotrowski
Ionut Cristian Pistol
Soujanya Poria
Luz Rello
Noushin Rezapour Asheghi
Francisco Ribadas-Pena
Alexandra Roshchina
Tobias Roth
Jan Rupnik
Upendra Sapkota
Gerold Schneider
Djamé Seddah
Keiji Shinzato
João Silva
Sara Silveira
Sen Soori

Sanja Stajner
Tadej Štajner
Zofia Stankiewicz
Hristo Tanev
Irina Temnikova
Mitja Trampus
Diana Trandabat
Yasushi Tsubota
Srinivas Vadrevu
Francisco Viveros Jiménez
Josh Weissbock
Clarissa Xavier
Victoria Yaneva
Manuela Yapomo
Hikaru Yokono
Taras Zagibalov
Vanni Zavarella
Alisa Zhila

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
noviembre de 2013
Printing 500 / Edición 500 ejemplares