Automatic Linguistic Pattern Identification Based on Graph Text Representation

Helena Gómez-Adorno[1], Grigori Sidorov[1], David Pinto[2],and Darnes Vilariño[2]

[1] Centro de Investigación en Computación,
Instituto Politécnico Nacional, Mexico
helena.adorno@gmail.com, sidorov@cic.ipn.mx
[2] Facultad de Ciencias de la Computación,
Benemérita Universidad Autonóma de Puebla, Mexico
{dpinto,darnes}@cs.buap.mx

Abstract. In this paper it is presented a model of text representation based on graphs. The model is applied in the particular case study of authorship attribution. The experiments were performed by using a corpus made up of 500 documents written by 10 different authors (50 documents per author). The obtained results highlight the benefit of using text features at different levels of language description in tasks associated to automatic processing of information. In particular, we have obtained a performance of 57% of accuracy for the authorship attribution task.

Key words: Graph representation, writing style patterns, authorship attribution.

1 Introduction

There is a huge amount of digital information produced daily in the form of natural language written texts, such as, magazines, books, web pages, newspapers, reports, etc. The exponential growth of these documents requires the development of new approaches in order to explore, analyse and discover knowledge from them. The development of new tools for discovering new, previously unknown information is one of the goals of the Text Mining (TM) research field. It is a non-trivial task to find accurate knowledge in text documents that are helpful to tackle a determined task. Some TM task are text categorization, text clustering, sentiment analysis, document summarization, authorship attribution, information retrieval, tagging/annotation, among others.

The representation of natural language text is needed in order to represent human knowledge in computers. Conventional text representation models observed in TM task are: Boolean models [1], probabilistic models [2] and vector space models [3]. The features used for text representation usually are Bag-Of-Words (BOW) [4] or n-grams models [5–7]. The majority of these text representations are based on the BoW representation, thus ignoring the words' sequentiality and, hence, the meaning implied or expressed in the documents as well. This deficiency generally results in failure to perceive contextual similarity

pp. 43–52; rec. 2014-02-20; acc. 2014-03-22

of text passages. This may be due to the variation of words that the passages contain. Another chance is perceiving contextually dissimilar text passages as being similar, because of the resemblance of their words.

For many problems in natural language processing, a graph structure is an intuitive, natural and direct way to represent the data. This paper presents an innovating graph-based framework for automatic analysis of texts, which was proposed in our previous work [8] for the specific task of document understanding. The new model will include linguistic features of different levels of language description, which provide important information about the knowledge that is expressed in a natural language written text.

The aim of the present research work is the construction of a suitable text representation model based on graphs, that can facilitate discovering of important text patterns from it. We show that the features (text patterns) in this manner discovered can be used in various tasks associated to document understanding (such as for document classification, information retrieval, information filtering, information extraction, authorship attribution).

The text pattern discovering technique proposed here is based on the traversal of the graph representation of documents, using the shortest paths. This text pattern discovery is used in our experimental case study for estimating similarities between pairs of texts. The case study of authorship attribution presented here demonstrates how our framework works and its' efficacy. The results of the experimental work reported here are analyzed, and current findings are discussed.

There exist several research works that have employed graphs for representing text. A comprehensive study of the use of graph-based algorithms for natural language processing and information retrieval can be found in [9]. It describes approaches and algorithmic formulations for: (a) synonym detection and automatic construction of semantic classes using measures of graph connectivity on graphs built from either raw text or user-contributed resources; (b) measures of semantic distance on semantic networks, including simple path-length algorithms and more complex random-walk methods; (c) textual entailment using graph-matching algorithms on syntactic or semantic graphs; (d) word-sense disambiguation and name disambiguation, including random-walk algorithms and semi-supervised methods using label propagation on graphs; and (e) sentiment classification using semi-supervised graph-based learning or prior subjectivity detection with min-cut/max-flow algorithms. Although the work described in [9] covers a wide range of algorithms and applications, there exist other relevant works in literature worth mentioning. A great interest has grown in the computational linguistic community for using this kind of text representation in diverse tasks of natural language processing, such as in summarization [10], coreference resolution [11], word sense disambiguation [12–14], word clustering [15, 16], document clustering [17], etc.

The majority of the approaches presented in literature use well known graph-based techniques in order to find and exploit the structural properties of the graph underlying a particular dataset. Because the graph is analysed as a whole, these techniques have the remarkable property of being able to find globally optimal solutions, given the relations between entities. For instance, graph-based methods are particularly suited for disambiguating word sequences, and they manage to exploit the interrelations among the senses in the given context. Unfortunately, most of the research works that use graph-based representations propose ad-hoc graph-structures that only work with the particular problem they are dealing with. It is, therefore, imperative to attempt to propose a general framework that may be used in different contexts with a minimum amount of changes.

In summary, this research work presents a new text representation schema useful for mining documents, exploiting their lexical, morpho-syntactic and semantic information. The representation schema is built over a syntactic analysis developed through a dependency parser for all the sentences in the document, including further morphologic and semantic information. The final result obtained is an enriched output in the form of a graph that represents the input document in the form of a multiple level formal representation of natural language sentences. The graph-based representation schema and the similarity measure proposed here, enables a more effective and efficient text mining process.

2 Research Methodology

The proposed text representation schema utilizes multiple linguistic levels of formal definition of natural language texts. The motivation for the schema is to capture most of the features present in a document, ranging from lexical to semantic level. By including lexical, syntactic, morphologic and semantic analysis in the representation, we attempt to represent how different text components (words, phrases, clauses, sentences, etc.) are related.

The representation of each linguistic level together with their association with the graph components is described as follows.

Lexical level At the lexical level we deal with words, one of the most basic units of text, describing their meaning in relation to the physical world or to abstract concepts, without reference to any sentence in which they may occur. Lexical definition attempts to capture everything that a term is used to refer to and, as such, is often too vague for many purposes.

Morphological level At the morphological level we deal with the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes and Parts of Speech (PoS).

Syntactical level At the syntactical level we deal with rules and principles that govern the sentence structures. Usually, the lexical parser (or simply: the parser) can read various forms of plain text input and can output various analysis formats, including part-of-speech tagged text (morphological level), phrase structure trees, and a grammatical relations (typed dependency) format.

Semantic level At the semantic level we deal with the meaning of sentence, i.e., human expression stated through language. In general, semantic level refers to interpretation of signs or symbols used in agents or communities within particular circumstances and contexts. In written language, things like paragraphs, words and punctuation symbols bear semantic content. The most popular semantic relationships are: antonym, synonym, class inclusion, part-whole, and case [18].

2.1 Formalization of the Graph-Based Multi-Level Linguistic Representation

Given a text $T = \{t_1, t_2, \cdots, t_{|T|}\}$ with t_i a word in the document. Let $PoS(t_i)$ be the PoS tag of t_i, $Lem(t_i)$ be the lemma of t_i, $Sem(t_i)$ be a term semantically related with t_i, and $Dep(t_i, t_k)$ be the dependency tag obtained by some syntactical parser over the sequence "$t_i t_k$". The graph-based multi-level linguistic representation of T can be formally expressed by a di-graph $G = \{V, E, L_V, L_E, \alpha, \beta\}$, with:

- $V = \{v_i | i = 1, ..., n\}$ is a finite set of vertices, $V \neq \emptyset$, and n is the number of vertices in the graph.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leq i, j \leq n\}$. Note that the notation (v_i, v_j) indicates that a given order is established.
- $L_V = \{\bigcup_{i=1,\cdots,|T|}(Lem(t_i) \bigcup Pos(t_i))\}$
- $L_E = \{\bigcup_{i,j=1,\cdots,|V|} Dep(v_i, v_j)$ with $v_i, v_j \in V,$ and $(v_i, v_j) \in E\}$
- $\alpha : V \rightarrow L_V$
- $\beta : E \rightarrow L_E$

Here, we say that L_E represents the dependency tag between a pair of words. However, it is more practical to have a numeric value as edge label in addition to the dependency tag. We, therefore, extend the graph-based representation using the following definition of L_E.

$L_E = \{\forall_{i,j=1,\cdots,|V|}(Dep(v_i, v_j) : frec(Dep(v_i, v_j)) + frec((v_i, v_j)))\}$
with $v_i, v_j \in V$, and $(v_i, v_j) \in E$

Where $frec(x)$ is a function that counts the occurrences of x in the entire graph.

Figure 1 shows the final constructed graph for an example sentence. To illustrate the graph representation, let us consider the following example: *Text mining searches patterns in texts.*

2.2 Pattern Extraction Technique for Discovering Text Features

The graph may represent one sentence, one paragraph, one document, or even a collection of documents. We assume that the graph uses the representations we discussed in the previous section. The technique proposes to find features in the graph by counting text components (word lemmas, PoS tags, grammatical tags) when different paths are traversed. These components would seem to be

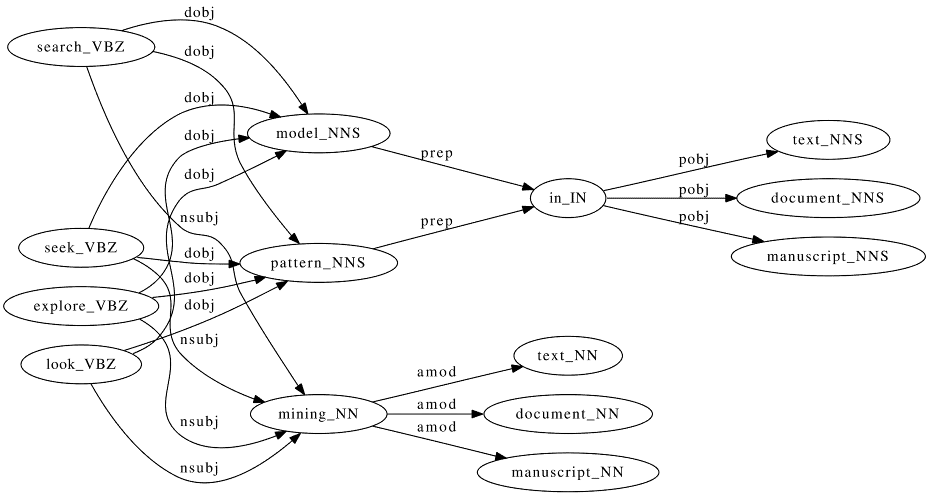

Fig. 1. Semantical representation of texts using word lemmas, PoS tags, dependency tags and word synonyms

isolated elements of the graph, however, counted over a path of interest they are considered to be textual patterns.

Let us consider the semantic representation shown in Figure 1, the minimum path from the node *search_VBZ* to the node *text_NNS* will have the following features at different language description levels:

– Lexical level: *search*, *model*, *text*, *in*.
– Morphological level: VBZ, NNS, IN, NNS.
– Syntactical level: *dobj*, *prep*, *pobj*.

Those features may be further used (perhaps as a bag of words or a vector space model based vector) for some particular task to be carried out. Thus, a textual document represented by a graph may provide a set of features for each of the minimum paths found in that graph. These features can be used for encoding a meta-representation of the text.

3 Case Study

In order to analyse the performance of the graph-based multi-level linguistic representation and the pattern extraction technique, we present their application in a particular problem of text mining known as Authorship Attribution.

The authorship attribution is often seen as an issue of computational linguistics and aims to identify the author or the original writer of a given text. To accomplish this task, it is necessary to identify characteristics (features) or profiles that identify the target author. This is not a trivial task because writing styles are often similar. Many different techniques have been develop in order to solve this problem, from mathematical and statistical applications to computational linguistic approaches.

The particular interest in this research field lies in the huge amount of existing documents and the need to develop automatic and reliable models that contribute to the optimal solution of the problem.

In order to determine the performance of the text representation proposed in this work in a real scenario, we used a corpus gathered by Efstathios Stamatatos(ccat C10) [19] from the RCV1 Reuters collection [20]. This corpus is made up of 500 documents (10 authors each one with 50 different documents) for both, the training and test dataset.

3.1 Applying the Proposed Methodology to the Authorship Attribution Task

By using the methodology proposed, we can construct ten graphs that include all the document collection. In this case, each graph will represent only one author (50 documents).

In order to identify the authorship of a given test document, we need first to represent the test document with the proposed graph-based representation 2. Thereafter, we use the pattern extraction technique introduced in Section 2.2 for obtaining numeric vectors and subsequently to calculate the similarity between the test graph and each of the ten author's graphs (which contains the graph-based representation of the fifty documents of each author). Thus, following the features extraction technique, the root node of the test graph is fixed as the initial node whereas the final nodes selected correspond to the rest nodes of the test graph. This leads to diminish the computational time to $O(n)$, with n equal to the number of nodes in the test graph. We have used the Dijkstra algorithm[21] for finding the minimum path between the initial and each final node. Thereafter, we count the occurrences of all the multi-level linguistic features considered in the text representation, such as part-of-speech tags and dependency tags found in the path. The same procedure is performed with the training graph by using the pair of words identified in test graph as initial and final nodes. As a result of this procedure, we obtain two set of feature vectors: one for the test graph, and one for the training graph.

The technique extracts a set of vectorial patterns $(\overrightarrow{f_{t,i}})$ for each text t, with V equal to the total number of lexical, morphological and syntactical features. Thus, the training graphs Tr will now be represented by m feature vectors $(Tr^* = \{\overrightarrow{f_{d,1}}, \overrightarrow{f_{d,2}}, \cdots, \overrightarrow{f_{d,m}}\})$, as well as the test graph Te $(Te^* = \{\overrightarrow{f_{h,1}}, \overrightarrow{f_{h,2}}, \cdots, \overrightarrow{f_{h,m}}\})$. Here, m is the number of different paths that may be traversed in both graphs, using the "ROOT-0" vertex as the initial node and each word appearing in the test graph as the final node.

Since each path of the test graph contains exactly the same number and types of components as that of the training graphs, it is possible to calculate the degree of similarity among each path traversed. For the purposes of this case study, we have used the cosine similarity measure, which is calculated as in Eq.(1).

$$
\begin{aligned}
Similarity(h^*, d^*) &= \sum_{i=1}^{m} Cosine(\overrightarrow{f_{h,i}}, \overrightarrow{f_{d,i}}) \\
&= \sum_{i=1}^{m} \frac{\overrightarrow{f_{h,i}} \cdot \overrightarrow{f_{d,i}}}{||\overrightarrow{f_{h,i}}|| \cdot ||\overrightarrow{f_{d,i}}||} \\
&= \sum_{i=1}^{m} \frac{\sum_{j=1}^{|V|} (f_{(h,i),j} * f_{(d,i),j})}{\sqrt{\sum_{j=1}^{|V|} (f_{(h,i),j})^2} * \sqrt{\sum_{j=1}^{|V|} (f_{(d,i),j})^2}} \qquad (1)
\end{aligned}
$$

After obtaining all the similarity scores for the test graph with respect to the ten training graphs, the training graph obtaining the highest score is selected as the correct author for that test graph.

3.2 Obtained Results

In Figure 2 we show the results obtained for each one of the ten authors. We have executed the following four different representation methods:

- *MinText_WithoutLem:* It represents the text in the graphs by using the document words (not lemmatized).
- *MinText_LemFrec:* It includes the word lemmatization plus the frequency count of the nodes in the entire graph.
- *MinText_LemFrecLexMor:* It includes the counting of lexical and morphological features.
- *MinText_LemFrecLexMorVocSuf:* It includes, besides all the previous features, the combination of vowels count, permutation of vowels count and suffix count.

As can be seen, there exist some authors that were benefited of using more complex components of text representation, for example, lemmatization and use of PoS tags. However, there are other authors who did not. We consider this behaviour is due to the coherency of writing style of some authors. However, this is an hypothesis that we need to analyse more into detail by reviewing each one of the documents written by each author. Some authors have a writing style that is much more easy to be discovered and modeled, such as “Alan Crosby”. In these cases, the inclusion of new characteristics did not contribute to a better classification accuracy because the simple use of their words is enough for detecting the writing style. For those authors that did not use a consistent writing style, the inclusion of new characteristics allowed to discover their possible textual patterns.

In Table 1, we present a summary of the results obtained by each representation model. In average, the *MinText_LemFrecLexMorVocSuf* model obtained the best performance.

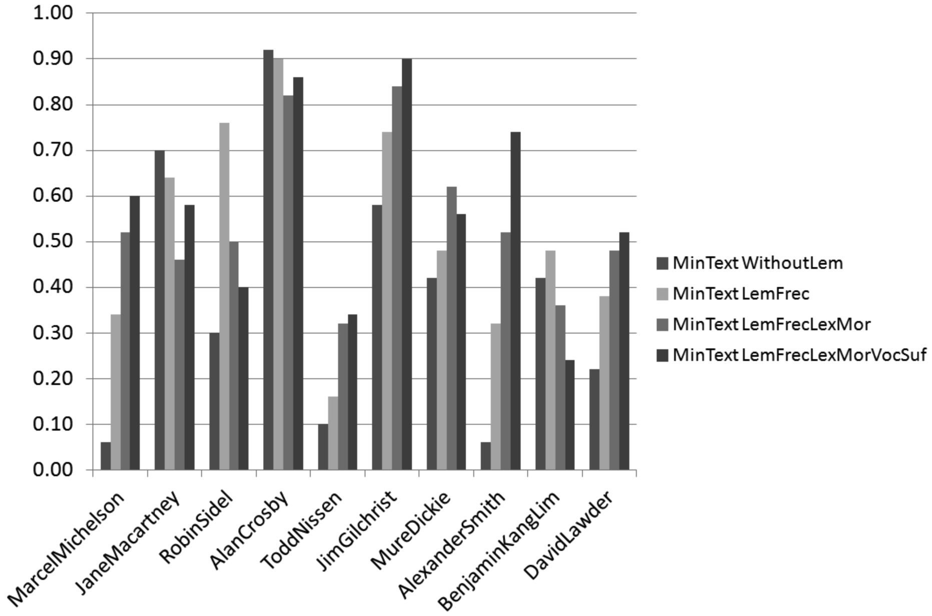

Fig. 2. Evaluation of accuracy for each one of the ten authors

Table 1. Comparison of the results obtained in the Authorship Attribution task

Evaluated approach	**2011**
MinText_WithoutLem	37.80
MinText_LemFrec	52.00
MinText_LemFrecLexMor	54.40
MinText_LemFrecLexMorVocSuf	**57.4**

4 Main Contribution

At the end of this research project we will count with a new methodological framework that can be used in various tasks of MT and PLN. In particular, the new methodology will consist of:

1. A model for text representation based on graph structures, using the different levels of natural language formal description.
2. A set of linguistic pattern extraction techniques useful for solving specific tasks of MT and PLN.
3. Metric for evaluating similarity between patterns discovered on different datasets.
4. Classification models for various PLN and MT tasks: text classification, authorship attribution and profiling, analysis of polarity, among others.

There exist, however, a gap in the process of selecting the best features to be included in the graph that will represent the target text, because, some features may help or harm the final task to be tackled. Despite this issue, we consider very important to propose a representation that takes into consideration features beyond the traditional ones (such as bag of words).

References

1. Mauldin, M.L.: Retrieval performance in ferret a conceptual information retrieval system. In: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '91, New York, NY, USA, ACM (1991) 347–355
2. Croft, W.B., Turtle, H.R., Lewis, D.D.: The use of phrases and structured queries in information retrieval. In: Proc. of the 14th SIGIR conference, New York, NY, USA, ACM (1991) 32–45
3. Salton, G., ed.: Automatic text processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1988)
4. Mladenic, D., Grobelnik, M.: Word sequences as features in text-learning. In: In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98). (1998) 145–148
5. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-based authorship attribution without lexical measures. In: Computers and the Humanities. (2001) 193–214
6. Keselj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution (2003)
7. Sidorov, G., ed.: Non-linear construction of n-grams in computational linguistics: syntactic, filtered, and generalized n-grams. Sociedad Mexicana de Inteligencia Artificial (2013)
8. Pinto, D., Gómez-Adorno, H., no, D.V., Singh, V.K.: A graph-based multi-level linguistic representation for document understanding. Pattern Recognition Letters **41**(0) (2014) 93 – 102
9. Mihalcea, R., Radev, D.: Graph-based natural language processing and information retrieval. Cambridge university press (2011)
10. Zha, H.: Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: SIGIR, ACM (2002) 113–120
11. Nicolae, C., Nicolae, G.: Bestcut: a graph algorithm for coreference resolution. In: Proc. of the EMNLP 2006 Conference, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 275–283
12. Dorow, B., Widdows, D.: Discovering corpus-specific word senses. In: EACL, The Association for Computer Linguistics (2003) 79–82
13. Veronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech & Language **18**(3) (2004) 223–252
14. Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A.: Two graph-based algorithms for state-of-the-art wsd. In Jurafsky, D., Gaussier, É., eds.: EMNLP, ACL (2006) 585–593
15. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based word clustering using a web search engine. In: Proc. of the EMNLP 2006 Conference, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 542–550
16. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Proc. of the 1st Workshop on Graph Based Methods for Natural Language Processing, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 73–80

17. Zhong, S.: Generative model-based document clustering: a comparative study. Knowledge and Information Systems **8** (2005) 374–384
18. Bejar, I., Chaffin, R., Embretson, S.: Cognitive and psychometric analysis of analogical problem solving. Recent research in psychology. Springer-Verlag (1991)
19. Houvardas, J., Stamatatos, E.: N-gram feature selection for authorship identification. In Euzenat, J., Domingue, J., eds.: Artificial Intelligence: Methodology, Systems, and Applications. Volume 4183 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2006) 77–86
20. Lewis, D.D., Yang, Y., Rose, T.G., Li, F., Dietterich, G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research **5** (2004) 361–397
21. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1) (1959) 269–271