Pattern Recognition Letters 41 (2014) 93–102

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A graph-based multi-level linguistic representation for document understanding ☆

David Pinto [a,*], Helena Gómez-Adorno [a,1], Darnes Vilariño [a,1], Vivek Kumar Singh [b,2]

[a] Benemérita Universidad Autonóma de Puebla, Faculty of Computer Science, 14 Sur & Av. San Claudio, CU, Edif. 104C, Puebla, Mexico
[b] South Asian University, Department of Computer Science, Akbar Bhawan, Chanakyapuri, New Delhi 110021, India

ARTICLE INFO

Article history:
Available online 9 December 2013

Keywords:
Text mining
Text representation
Graph-based representation

ABSTRACT

Document understanding goal requires discovery of meaningful patterns in text, which in turn requires analyzing documents and extracting information useful for a purpose. The documents to be analyzed are expected to be represented in some way. It is true that different representations of the same piece of text might have different information extraction outcomes. Therefore, it is very important to propose a reliable text representation schema that may incorporate as many features as possible, and at the same time provides use of efficient document understanding algorithms. In this paper, we propose a graph-based representation of textual documents that employs different levels of formal representation of natural language. This schema takes into account different linguistic levels, such as lexical, morphological, syntactical and semantics. The representation schema proposed is accompanied with a proposal for a technique which allows to extract useful text patterns based on the idea of minimum paths in the graph. The efficiency of the representation schema proposed has been tested in one case of study (Question-Answering for machine Reading Evaluation – QA4MRE), and the results of experiments carried in it, are described. The results obtained show that the proposed graph-based multi-level linguistic representation schema may be successfully used in the broader framework of document understanding.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

A huge amount of information produced on a daily basis is found in different forms of natural language written texts, such as magazines, books, e-books, journals, technical reports, etc. In fact, we are now overwhelmed with textual data, which increases every other day. The explosive growth in the number of such documents needs development of effective approaches to explore, analyze, and discover knowledge from documents. Developing automated tools for machine reading by discovering patterns and extracting knowledge from texts is one of the most important goals of Text Mining (TM) research. And the usual assumption in it is that texts are represented in some kind of structure.

The present research work is mainly concerned with the construction of a suitable text representation model based on graphs, that can facilitate discovering of important text patterns from it. We propose to state and demonstrate that the features (text patterns) so discovered can be used in different tasks associated to document understanding (such as for document classification, information retrieval, information filtering, information extraction and question answering).

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 question answering validation for reading comprehension tests presented here demonstrates the working and efficacy of our framework. The results of experimental work reported are analyzed and key observations clearly stated.

In summary, this research work presents a new text representation schema useful for mining documents, exploiting their lexical, syntactic, morphologic 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.

The rest of the paper is organized as follows. Section 2 presents a literature survey on the different text representation schemata

☆ This paper has been recommended for acceptance by J. Fco. Martínez-Trinidad.

* Corresponding author. Tel.: +52 222 2295500x2856.
E-mail addresses: dpinto@cs.buap.mx (D. Pinto), helena.adorno@gmail.com (H. Gómez-Adorno), darnes@cs.buap.mx (D. Vilariño), vivekks12@gmail.com (V.K. Singh).
[1] Tel.: +52 222 2295500x2856.
[2] Tel.: +91 11 24195148.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.12.004

proposed. It also emphasizes the contribution of using graph-based structures in the text representation research field. Section 3 explains in detail the graph-based text representation schema proposed. The diverse features that may be included into this representation are discussed along with suitable examples. Section 4 describes our proposal of an efficient method for discovering texts patterns from the graph-based representation of text documents. Section 5 presents the performance assessment of the proposed schema of text representation, in the particular case study of QA4MRE. It first describes the task and then illustrates the process of discovering text patterns. Finally, the results obtained in the experiments are reported. Section 6 concludes the paper by presenting the main contribution and findings of this research work.

2. State of the art

The most conventional text representation schemata observed in applications like information retrieval, text categorization, authorship attribution etc. are: Bag of Words (BoW) (Mladenic and Grobelnik, 1998), *n*-grams model (Stamatatos et al., 2001; Keselj et al., 2003), boolean models (Mauldin, 1991), probabilistic models (Croft et al., 1991) and vector-space models (Salton, 1988). 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 of text passages. This may be due to the variation of words that the passages contain. Another possibility 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. 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 Mihalcea and Radev (2011). 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 Mihalcea and Radev (2011) covers a wide number 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 (Zha, 2002), coreference resolution (Nicolae and Nicolae, 2006), word sense disambiguation (Dorow and Widdows, 2003; Veronis, 2004; Agirre et al., 2006), word clustering (Matsuo et al., 2006; Biemann, 2006), document clustering (Zhong, 2005), 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 analyzed 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.

3. A graph-based multi-level linguistic representation schema for documents

This section presents our proposed text representation schema that 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.

A labeled di-graph denoted by $G = \{V, E, L_V, L_E, \alpha, \beta\}$ is the starting point for representing the different levels of language description. Here:

- $V = \{v_i | i = 1, \ldots, n\}$ is a finite set of vertices, $V \neq \emptyset$, and n is the number of vertices in the graph.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leqslant i,j \leqslant n\}$. Note that the notation (v_i, v_j) indicates that a given order is established.
- L_V is the tag set for the vertices.
- L_E is the tag set for the edges.
- $\alpha : V \rightarrow L_V$ is a function that assigns tags to vertices.
- $\beta : E \rightarrow L_E$ is a function that assigns tags to the directed edges.

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

3.1. 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. Therefore, it is used as a basic representation which need to be further enriched through higher levels of language description.

To illustrate the lexical level of representation, let us consider the following example sentence:

Text mining searches patterns in texts.

Thus, given a di-graph $G = \{V, E, L_V, L_E, \alpha, \beta\}$, the function α assigns lexical words to the vertices. In this case, the L_V set (set of all the lexical words found in the document to be represented) is $L_V = \{$"*Text*", "*mining*", "*searches*", "*patterns*", "*in*", "*texts*"$\}$. At this point, we have only assigned lexical components to the vertices of the graph, thus, the edges are not defined yet. In other words, there are no edges to reflect any relationship among the words in the graph. This is a basic representation that it is barely useful for practical purposes. Therefore, we move ahead to capture and represent the morphological level details of the language description.

3.2. 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). In order to introduce these morphological

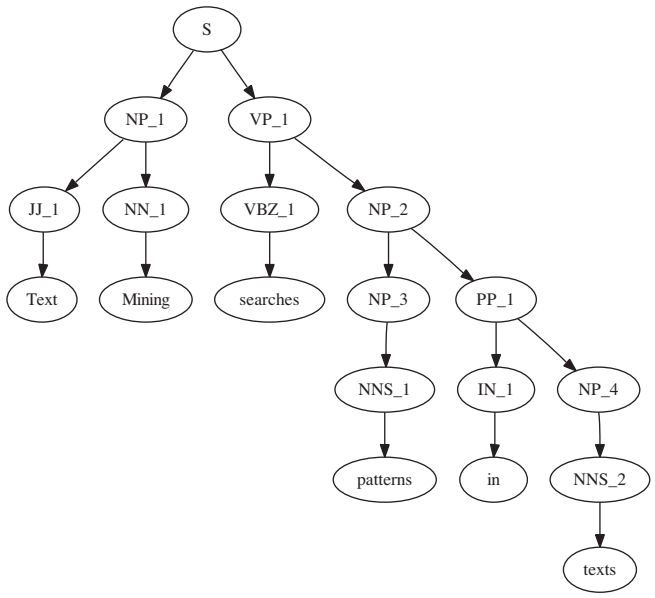


Fig. 1. Phrase parsing of the sentence: "Text mining searches patterns in texts".

components into our proposed representation, we have obtained the PoS tags using the Stanford Log-linear Part-Of-Speech Tagger.[3] The word lemmas were obtained using the TreeTagger.[4] It would be in order to mention here that the Penn Treebank tag set (Marcus et al., 1993) used in the morphological analysis of the texts contains 36 POS tags and 12 other tags (for punctuation and currency symbols).

For the example sentence mentioned in the previous section, we include a second level of language description in the graph-based representation by considering both, PoS tags and the word lemmas in the graph vertices. Thus, $L_V = \{$"*text_NN*", "*mining_NN*", "*search_VBZ*", "*pattern_NNS*", "*in_IN*", "*text_NNS*"$\}$. Note that this representation does not consider the original words anymore, since they have been replaced with the corresponding lemmas.

3.3. 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. Different syntactic-based parsers exist in literature, however, for the purposes of this work, we have shown the output generated by the Stanford parser.[5]

Fig. 1 shows the phrase structure tree for the example sentence considered in the previous subsections. The tree structure starts from a phrasal label *S* which means "Sentence", followed by other phrasal labels such as *NP* (Noun Phrase), *VP* (Verbal Phrase) or *PP* (Prepositional Phrase). The last level of the tree contains the PoS tags followed by the word tagged. The definitions make use of the Penn Treebank part-of-speech tags and phrasal labels.[6] This type of parsing maintains the sequence of the words in the sentence. It may thus be used for enriching the representation with the parsing tags, but word dependencies are still not discovered.

In Fig. 2, we see another type of parsing (grammatical relations or typed dependency) applied to the same text of example. The dependencies are all binary relations: a grammatical relation holds between a governor (also known as a regent or a head) and a dependent. The description of the Stanford tags used in this paper are given in catherine De Marneffe and Manning (2008). In this type of parsing, we may take advantage of the grammatical relation obtained between two components of the sentence. With respect to the phrasal parsing, this representation is more compact, as it will be seen in the next subsection. It is more flexible for adding higher level language description levels, such as the semantic one.

3.4. 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 paragraph structure, word usage and punctuation bear semantic content. There exist several papers in literature approaching the linguistic semantics area, however, in this paper we are particularly interested in semantic relationships. A number of semantic relationships have been identified by researchers in different disciplines such as linguistics, logic, and cognitive psychology (Storey, 1993). The most popular semantic relationships are: antonym, synonym, class inclusion, part-whole, and case. Semantic relationships, together with a description of them, have been proposed in the work developed by Bejar et al. (1991).

Fig. 3 shows the manner we can integrate "synonyms" in the graph-based representation, however, other semantic relationships could be also included in the graph. For instance, the vertex "text_NN" is expanded with two synonyms: "document_NN" and "manuscript_NN", which are then linked to the same vertices in the graph corresponding to the original node "text_NN" (edge direction is kept).

3.5. 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:

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

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 $(v_i, v_j) \in E$

[3] <http://nlp.stanford.edu/software/tagger.shtml>.
[4] <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.
[5] <http://nlp.stanford.edu/software/lex-parser.shtml>.
[6] <http://www.cis.upenn.edu/treebank/>.

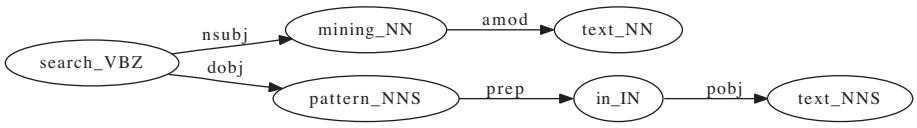


Fig. 2. Syntactical representation of texts using word lemmas, PoS tags and dependency tags.

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

Thus, each edge contains the dependency tag together with a number that indicates the frequency of that dependency tag plus the frequency of the pair of vertices, both calculated over the complete graph. Fig. 4 depicts the graph that considers the labeling extension in the graph edges. The figure shows the representation for the same example discussed throughout this paper. In this figure, we have added the numbers associated to the frequency of the dependency tag and the frequency of the edge between two given vertices as well. This have been done for descriptive purposes, however, in the final representation, those values are not stored in the graph, but only the sum of the two values. For instance, the edge between the vertices "mining_NN" and "text_NN" has been labeled as "amod:4", which means that "amod" is the dependency tag that exists between these two vertices. Additionally, the number 4 means that the "amod" dependency tag appears 3 times in the graph and the frequency of the pair ("mining_NN", "text_NN") in the graph is 1, thus 4 = 3 + 1.

4. MinText: a feature extraction technique for discovering text patterns

In this section we present a feature extraction technique for finding patterns in graph representation of a given text. 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 MinText 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 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 Fig. 3, 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.

In Table 1 we can see an example of the features extracted with minimum paths, in which each row represents one path.The number of pairs considered as initial and final node may vary, for instance, considering all the combinations for the n nodes in the graph (the complexity time will be $O(n^2)$), or fixing the initial or the final node (the complexity time will be $O(n)$). Different decisions can be made based on the particular mining text task to be accomplished.

The MinText technique takes advantage of the different linguistic description levels represented in the graph. It codifies textual information to numeric values which may be further used, for instance, to feed machine learning methods, or to calculate textual similarities among different texts.

5. Case study: QA4MRE

In order to analyze the performance of the graph-based multilevel linguistic representation and the MinText technique, we present their application in a particular problem of document

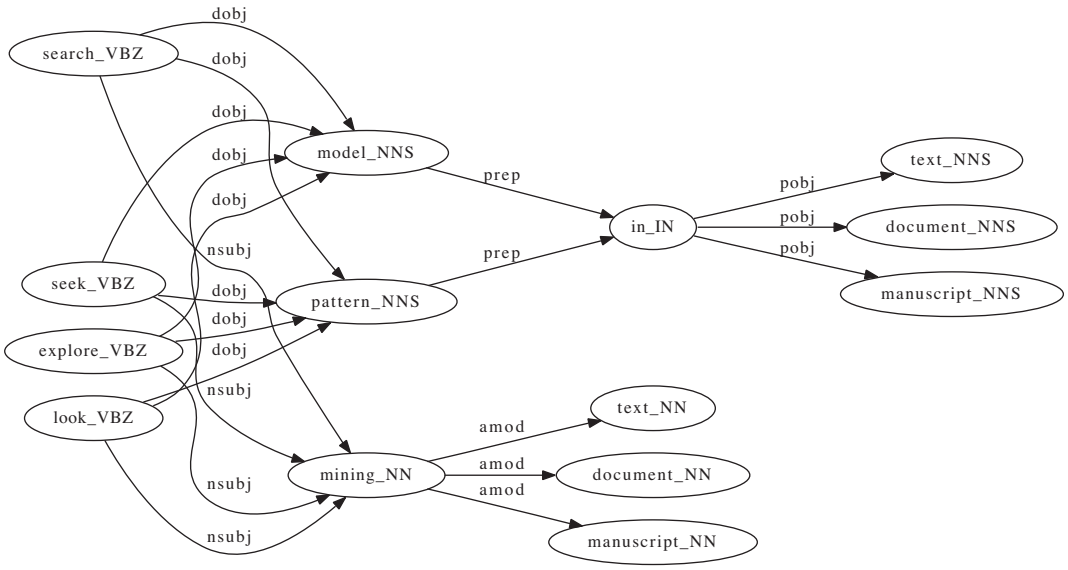


Fig. 3. Semantical representation of texts using word lemmas, PoS tags, dependency tags and word synonyms.

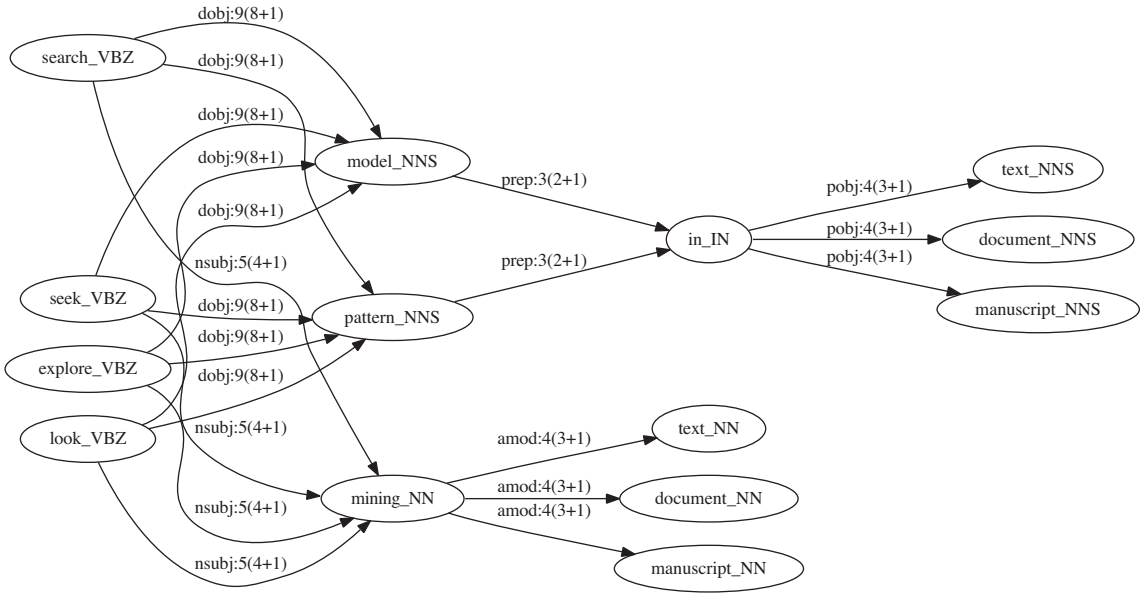


Fig. 4. Graph-based representation with numeric values in the edges.

understanding known as "Question Answering for Machine Reading Evaluation (QA4MRE)". The details of implementation of both, the representation schema and the MinText technique, are described below. We also illustrate the case study corpora of evaluation and experimental results.

5.1. Task description

The QA4MRE task was first proposed in the 2011 edition of the CLEF conference.[7] The main objective of this task has been to develop a methodology for evaluating Machine Reading systems through Question Answering and Reading Comprehension Tests. Systems to be evaluated should be able to extract knowledge from large volumes of text and use this knowledge to answer questions.

The task focuses on the reading of single documents and the identification of answers to questions about information that is stated or implied in the text. Systems should be able to use knowledge obtained automatically from input texts in order to answer a set of questions posed for single documents at a time.

5.2. Corpus description

In order to determine the performance of the text representation proposed in this paper in a real scenario, we used the corpora provided in the QA4MRE task of the CLEF 2011 and 2012. Even though the two datasets look similar at first glance, in practice they produce different results in the systems developed. Most of the questions of the corpus provided in 2011 have been written in first person, thus leading to obtain better performance if the system takes this issue into consideration. The first dataset (CLEF 2011) contains the following three topics: Climate Change, Music & Society and AIDS. The second dataset (CLEF 2012) contains four topics: Climate Change, Music & Society, Alzheimer and AIDS. Both datasets provide 10 questions for each one of the 4 reading tests given per topic. Therefore, the total number of questions is 120 for the first corpus (2011), whereas there are 160 questions for the second one. Each question has 5 multiple-choice answers from which only one answer must be selected as the correct one. For a complete description of these datasets (see Peñas et al., 2011, 2012).

5.3. Applying the proposed representation and the MinText technique to QA4MRE

The QA4MRE task aims to select the correct answer for a given question, using only one small document (≈500–1000 words) as reference. The approach proposed considers to formulate candidate answers (named "answer hypothesis") which are then validated in order to determine the one that best matches with respect to the document of reference. These candidate answers are an improved version of the original question, removing some cue words associated to the questions, such as *who*, *where*, *which*, and replacing these cue words with one of the possible answers given in the test.

Let us consider the following test (question-answers):

Question 1: **Who** is the founder of the SING campaign?
Answer 1: Nelson Mandela
Answer 2: Youssou N'Dour
Answer 3: Michel Sidibe
Answer 4: Zackie Achmat
Answer 5: Annie Lennox

Therefore, we can construct five different answer hypothesis as follows:

Hypothesis 1: **Nelson Mandela** is the founder of the SING campaign
Hypothesis 2: **Youssou N'Dour** is the founder of the SING campaign
Hypothesis 3: **Michel Sidibe** is the founder of the SING campaign
Hypothesis 4: **Zackie Achmat** is the founder of the SING campaign
Hypothesis 5: **Annie Lennox** is the founder of the SING campaign

We can validate each one of these answer hypothesis by comparing its similarity with respect to the reference document. In or-

[7] Conference and Labs of the Evaluation Forum; http://www.clef-initiative.eu/.

Table 1
Representation of a text using the MinText technique.

Initial node to final node	Lexical features				Morphological features				Syntactical features			
	search	model	···	text	NN	NNS	···	VBZ	dobj	prep	···	pobj
search_VBZ to text_NNS	1	1	···	1	0	2	···	1	1	1	···	1
search_VBZ to document_NNS	1	1	···	0	0	2	···	1	1	1	···	1
search_VBZ to in_IN	1	1	···	0	0	1	···	1	1	1	···	0
⋮	⋮				⋮				⋮			
look_VBZ to manuscript_NN	0	0	···	0	2	0	···	1	0	0	···	0

der to do so, we propose to represent both, the answer hypothesis and the reference document using the graph-based multi-level linguistic representation presented in Section 3.

Thereafter, we can use the MinText technique introduced in Section 4 for obtaining numeric vectors and subsequently to calculate the similarity between the reference document and each of the hypotheses. The hypothesis that obtain the highest score will be the one that will be selected as the correct answer to the question given.

The construction process and the validation procedure for the answer hypotheses is described below.

The hypotheses generator module receives as input the question set with their corresponding multiple-choice answers. As mentioned before, each hypothesis is constructed as the concatenation of the question with each of the possible answers. In order to generate the hypothesis, the "question keyword" is identified first, and afterwards it is replaced with each one of the five possible answers, thereby obtaining five hypotheses for each question. This hypothesis is intended to become the input of the Answer Validation (AV) module. The benefit of using these hypotheses as queries for the AV module is to search passages containing words that are in both, the question and the multiple-choice answer, instead of searching passages containing words from the question and the answer, independently.

5.4. Answer Validation

The Answer Validation module aims to assign a score to each hypothesis generated in the *Hypothesis generator* module. The Hypothesis obtaining the highest score is selected as the correct answer to the question.

Text documents along with its hypotheses are parsed to produce their lexical, morphological, syntactic and semantic graph representation (as described in Section 3). As a result of this process, each document is represented as a tree with branches to sub-trees that represent all the sentences in the document. The nodes of the tree represent the word lemmas of the sentences along with its part-of-speech tag. The branches represent the dependency tag between the two connecting nodes. In the same way the hypotheses are represented as a tree with the same characteristics as well.

In Fig. 5 we show the graph-based representation for two hypotheses considered in this case study: "Annie Lennox is the founder of the SING campaign" and "Nelson Mandela is the founder of the SING campaign"; whereas, Fig. 6 shows the graph-based representation for the first sentences of the reference document associated to the given question.

In order to extract the features and measure the similarity between a hypothesis and the reference document, the MinText technique is employed. The root node of the hypothesis graph is fixed as the initial node in the MinText technique, whereas the final nodes selected correspond to the rest nodes of the hypothesis graph. This leads to diminish the computational time to $O(n)$, with n equal to the number of nodes in the hypothesis graph. We have used the Dijkstra algorithm (Dijkstra, 1959) for finding the minimum path between the initial and each final node. Thereafter, following the MinText technique, we count the occurrences of all the multi-level linguistic features considered in the text representation, such as part-of-speech tags and dependencies tags found in the path. The same procedure is performed with the document graph by using the pair of words identified in the hypothesis as initial and final nodes. As a result of this procedure, we obtain two set of feature vectors: one for the answer hypothesis, and one for the reference document.

For instance, the minimum path between the initial node "root_0" and the final node "SING_NNP" calculated over the reference document (in Fig. 6) is "root_0" → "name_NN" → "of_IN" → "campaign_NN" → "Campaign_NNP" → "SING_NNP". In this path we can find two "NN" tags, one "IN" tag and two "NNP" tags. However, the number of "NNP" tags extracted from the same path using the graph presented in Fig. 5(a) is only one. Table 2 shows a partial view of the feature set for both, the correct answer hypothesis and the reference document, whereas, Table 3 shows a partial view of the feature set for both, an incorrect answer hypothesis and the reference document. Even if the reference document is the same for both answer hypothesis, the feature set for this document will change because these features are calculated taken as input the pair of nodes found in the corresponding answer hypothesis.

The MinText technique extracts a set of vectorial features ($\overrightarrow{f_{t,i}}$) for each text t, with V equal to the total number of lexical, morphological and syntactical features. Thus, the reference document d will now be represented by m feature vectors ($d^* = \{\overrightarrow{f_{d,1}}, \overrightarrow{f_{d,2}}, \ldots, \overrightarrow{f_{d,m}}\}$), as well as the answer hypothesis h ($h^* = \{\overrightarrow{f_{h,1}}, \overrightarrow{f_{h,2}}, \ldots, \overrightarrow{f_{h,m}}\}$). 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 hypothesis as the final node.

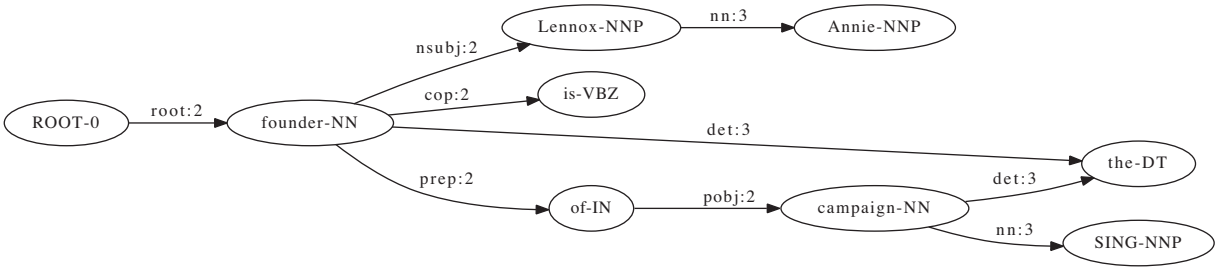
Since each path of the answer hypothesis contains exactly the same number and types of components as that of the reference document, it is possible to calculate the degree of similarity among each path traversed. For the purposes of this case study, we have used the cosine similarity measure, which is calculated as in Eq. (3).

$$Similarity(h^*, d^*) = \sum_{i=1}^{m} Cosine(\overrightarrow{f_{h,i}}, \overrightarrow{f_{d,i}}) \quad (1)$$

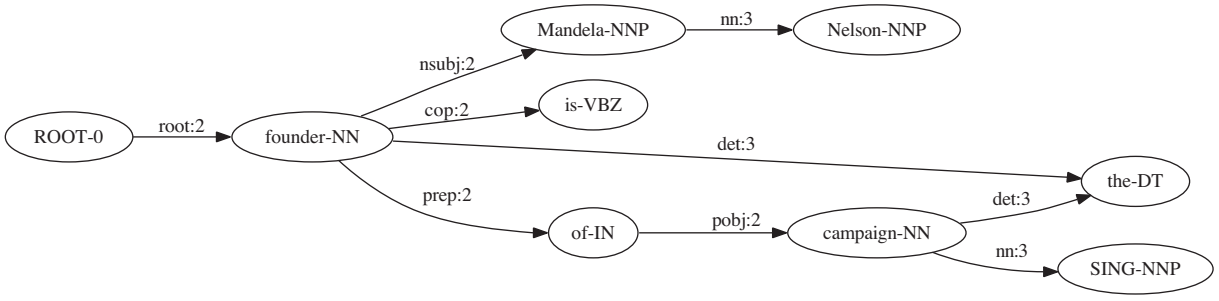
$$= \sum_{i=1}^{m} \frac{\overrightarrow{f_{h,i}} \cdot \overrightarrow{f_{d,i}}}{||\overrightarrow{f_{h,i}}|| \cdot ||\overrightarrow{f_{d,i}}||} \quad (2)$$

$$= \sum_{i=1}^{m} \frac{\sum_{j=1}^{|V|} (f_{(h,i),j} * f_{(d,i),j})}{\sqrt{\sum_{j=1}^{|V|} (f_{(h,i),j})^2} * \sqrt{\sum_{j=1}^{|V|} (f_{(d,i),j})^2}} \quad (3)$$

If some path does not exist in the reference document, then the feature vector will have zero values in all the feature weights, which will lead to an undefined equation. In this particular case, we have considered that the similarity between the two feature vectors will be equal to zero.



a) "Annie Lennox is the founder of the SING campaign"



b) "Nelson Mandela is the founder of the SING campaign"

Fig. 5. Graph-based representation of two different hypotheses.

After obtaining all the similarity scores for the five hypotheses of one question, the hypothesis obtaining the highest score is selected as the correct answer.

For instance, let us consider the first pair ($i = 1$) presented in Table 2, which corresponds to the path "root_0" to "Annie_NNP". The following equations show the manner the cosine similarity between $f_{h,1}$ and $f_{d,1}$ is calculated.

$$\overrightarrow{f_{h,1}} \cdot \overrightarrow{f_{d,1}} = 1 * 0 + 1 * 1 + \cdots + 0 * 0 + 1 * 1 + 0 * 0 + \cdots + 2 * 2 + 1 * 1 + 0 * 0 + \cdots + 0 * 0$$

$$||\overrightarrow{f_{h,1}}|| = \sqrt{(1^2 + 1^2 + \cdots + 0^2 + 1^2 + 0^2 + \cdots + 2^2 + 1^2 + 0^2 + \cdots + 0^2)}$$

$$||\overrightarrow{f_{d,1}}|| = \sqrt{(0^2 + 1^2 + \cdots + 0^2 + 1^2 + 0^2 + \cdots + 2^2 + 1^2 + 0^2 + \cdots + 0^2)}$$

$$Cosine(f_{h,1}, f_{d,1}) = \frac{\overrightarrow{f_{h,1}} \cdot \overrightarrow{f_{d,1}}}{||\overrightarrow{f_{h,1}}|| * ||\overrightarrow{f_{h,1}}||} \approx 0.93 \quad (4)$$

The same procedure is carried out with the remaining pairs. Therefore, the final score is obtained by adding the cosine score calculated with each pair.

On the other hand, considering the first pair ($i = 1$) presented in Table 3, which corresponds to the path "root_0" to "Nelson_NNP". This path does not exist in the graph representation of the reference document, therefore, all the features have zero values leading to an undefined equation. In this particular case, we have considered that the cosine similarity between the two feature vectors is equal to zero.

$$\overrightarrow{f_{h,1}} \cdot \overrightarrow{f_{d,1}} = 1 * 0 + 1 * 0 + \cdots + 0 * 0 + 1 * 0 + 0 * 0 + \cdots + 2 * 0 + 1 * 0 + 0 * 0 + \cdots + 0 * 0 = 0$$

$$||\overrightarrow{f_{h,1}}|| = \sqrt{(1^2 + 1^2 + \cdots + 0^2 + 1^2 + 0^2 + \cdots + 2^2 + 1^2 + 0^2 + \cdots + 0^2)}$$

$$||\overrightarrow{f_{d,1}}|| = \sqrt{(0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2 + \cdots + 0^2 + 0^2 + 0^2 + \cdots + 0^2)} = 0$$

$$Cosine(f_{h,1}, f_{d,1}) = 0 \quad (5)$$

Again, the same procedure is carried out with the remaining pairs. Therefore, the final score is obtained by adding the cosine score calculated with each pair.

In summary, if we consider the hypothesis "Annie Lennox is the founder of the SING campaign" (h_1), the following compute against the reference document (d) has to be done:

$Similarity(h_1, d) = Cosine$("root_0 to Annie_NNP", d) + $Cosine$("root_0 to is_VBZ", d) + $\cdots$ + $Cosine$("root_0 to SING_NNP", d) ≈ 2.59

Whereas, considering the hypothesis "Nelson Mandela is the founder of the SING campaign" (h_2), the following compute has to be done:

$Similarity(h_2, d) = Cosine$("root_0 to Nelson_NNP", d) + $Cosine$("root_0 to is_VBZ", d) + $\cdots$ + $Cosine$("root_0 to SING_NNP", d) ≈ 1.61.

Here we can see the evidence of the higher degree of similarity between the correct answer hypothesis (h_1) and the reference document (d) than the one calculated using the incorrect answer hypothesis (h_2).

5.5. Experimental results and evaluation

For the evaluation procedure we have used the $c@1$ measure, defined in Eq. (6). This measure was defined in the QA4MRE task of CLEF 2011 with the purpose of allowing to decide whether or not to answer a given question, i.e., the possibility of having questions with no answer. The aim of this measure is thus to reduce the amount of incorrect answers, maintaining the number of correct ones.

$$c@1 = \frac{1}{n}\left(n_R + n_U \frac{n_R}{n}\right) \quad (6)$$

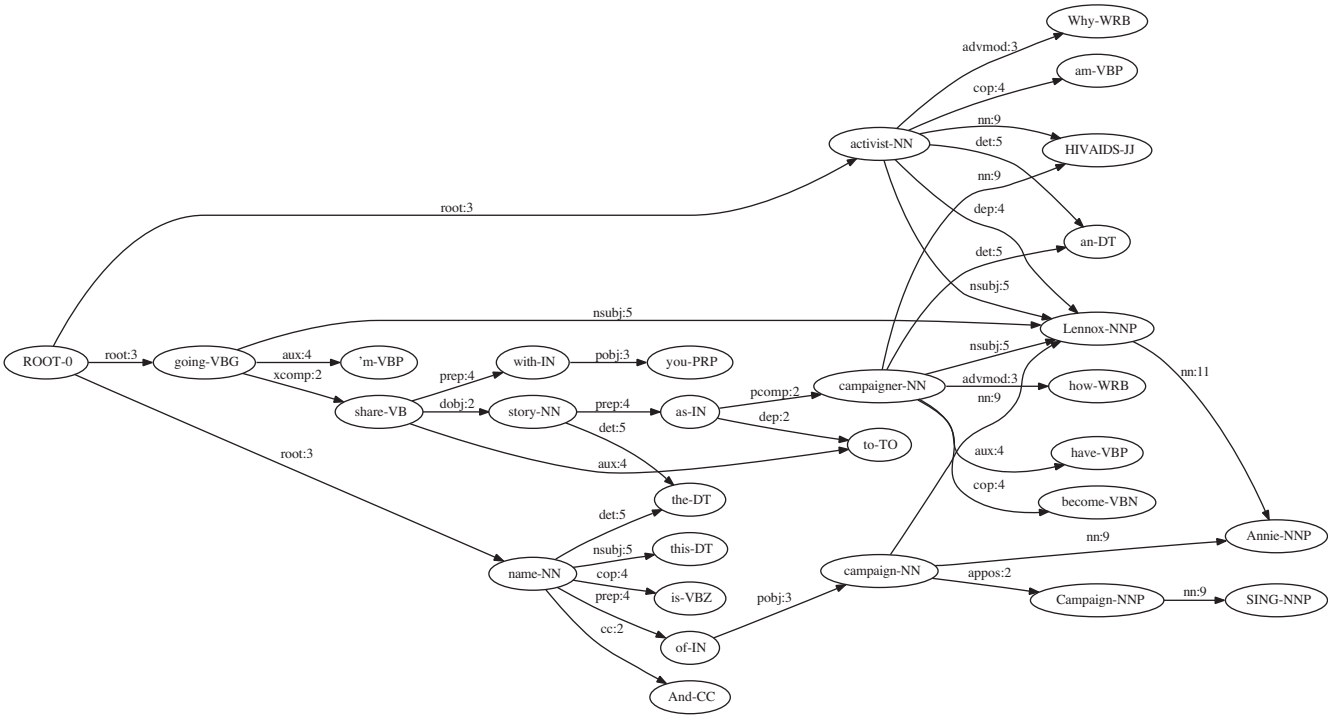


Fig. 6. Graph-based representation for one reference document.

Table 2
Representation of the answer hypothesis and the reference text (Answer hypothesis: "Annie Lennox is the founder of the SING campaign").

Initial node to final node	Lexical features				Morphological features				Syntactical features			
	founder	Lennox	···	campaign	NN	IN	···	NNP	nsubj	prep	···	pobj
Features extracted from the answer hypothesis graph-based representation												
root_0 to Annie_NNP	1	1	···	0	1	0	···	2	1	0	···	0
root_0 to is_VBZ	1	0	···	0	1	0	···	0	0	0	···	0
⋮	⋮				⋮				⋮			
root_0 to SING_NNP	1	0	···	1	2	1	···	1	0	1	···	1
Features extracted from the reference document graph-based representation												
root_0 to Annie_NNP	0	1	···	0	1	0	···	2	1	0	···	0
root_0 to is_VBZ	0	0	···	0	1	0	···	0	0	0	···	0
⋮	⋮				⋮				⋮			
root_0 to SING_NNP	0	0	···	1	2	1	···	2	0	1	···	1

where:

n_R: number of correctly answered questions
n_U: number of unanswered questions
n: total number of questions

Table 4 presents the results obtained with different approaches. The approach *MinText_TreeTagger* considers the application of the MinText technique when the words in the graph are lemmatized using the "TreeTagger" part of speech tagger. *MinText_Lancaster* is the one that uses the Lancaster stemmer. *MinText_Synonym* is an approach that expands each word with its corresponding synonyms (without applying the process of word sense disambiguation). *MinText_Hyponym* expands each word with its corresponding set of hyponyms. It is worth mentioning that all these approaches are implemented exclusively using basic techniques, but other variations may be suggested, for instance, considering a more complex analyses of the type of questions, or adding knowledge extracted from lexical or semantical resources such as ontologies.

As an additional analysis, we have carried out experiments towards the solution of the QA4MRE task without the use of the proposed MinText technique. The aim is to evaluate whether or not, the MinText is useful for extracting meaningful features from the graph-based representation. The results obtained with this implementation are also shown in Table 4, with the label of *Without_MinText*. Considering that the QA4MRE task requires to answer a question associated to the understanding of a given text, we have used the graph representation presented in Section 3 for both, the question (actually, the hypothesis of the question) and the document. Thereafter, we search the graph of the hypothesis in the document graph by means of partial matching. As a similarity measure, we count those edges that are in both graphs, using only the intersection of vertices between the two graphs (n) divided by the total number of posible edges $\left(\frac{n*(n-1)}{2}\right)$. This exercise was performed, as mentioned before, for determining the contribution of the MinText technique. We can observe that the use of the *Without_MinText* technique obtains a performance below the

Table 3
Representation of the answer hypothesis and the reference text (Answer hypothesis: "Nelson Mandela is the founder of the SING campaign").

Initial node to final node	Lexical features				Morphological features				Syntactical features			
	founder	Nelson	···	campaign	NN	IN	···	NNP	nsubj	prep	···	pobj
Features extracted from the answer hypothesis graph-based representation												
root_0 to Nelson_NNP	1	1	···	0	1	0	···	2	1	0	···	0
root_0 to is_VBZ	1	0	···	0	1	0	···	0	0	0	···	0
⋮	⋮				⋮				⋮			
root_0 to SING_NNP	1	0	···	1	2	1	···	1	0	1	···	1
Features extracted from the reference document graph-based representation												
root_0 to Nelson_NNP	0	0	···	0	0	0	···	0	0	0	···	0
root_0 to is_VBZ	0	0	···	0	1	0	···	0	0	0	···	0
⋮	⋮				⋮				⋮			
root_0 to SING_NNP	0	0	···	1	2	1	···	2	0	1	···	1

Table 4
A comparison of the results obtained in the QA4MRE task (English language).

Evaluated approach	2011	2012
MinText_TreeTagger	0.40	0.24
MinText_Lancaster	**0.42**	0.24
MinText_Synonym	0.37	0.23
MinText_Hyponym	0.36	**0.27**
Without_MinText	0.12	0.18
State-of-the-art techniques		
Best result	0.57	0.65
Avg. over all best runs	0.28	0.32
Avg. over all runs	0.21	0.26
Random baseline	0.20	0.20
Worst result	0.02	0.14

baseline for both datasets. This result indicates that the MinText alone is capable of extracting much more meaningful features than some other techniques that attempts to find the hypothesis (represented as a graph) directly in the document graph-based representation.

With respect to the comparison with the state-of-the-art, the *MinText* technique obtains a third place using the 2011 dataset with a c@1 of 0.42; a performance below two runs submitted by the same authors which obtained a c@1 of 0.47 and 0.57 (see Peñas et al., 2011). When we executed the same approach using the 2012 dataset, our performance was much more lower since it achieved a c@1 of 0.27, which rank us in the 11th place (see Peñas et al., 2012). Those results may seem discouraging, however, a fast review of the other runs of the state-of-the-art indicates a possibility of other techniques that can be employed for improving the obtained results. Textual entailment judgment, named entity recognition, type of question analysis, are some of these NLP techniques that will surely improve the final results of the QA4MRE task. We consider that the use of domain-specific techniques of natural language processing should improve the performance of the algorithm for this particular problem. However, the aim of this work is to show the graph-based representation and propose a technique to extract features from the graph, rather than solve the case study in an optimal manner. To maintain the simplicity and logical clarity of our description, we have not tried to include description of other domain specific techniques.

Nevertheless, it can be observed in Table 4 that in the QA4MRE 2011 dataset, the average over all best runs and over all runs were exceeded. As we have just outperformed the baseline of the QA4MRE dataset, we consider that the second dataset have been constructed using other type of language phenomena in the test questions and the reference documents such as anaphoric or cataphoric expressions. Additionally, we have frequently found negations of questions in the second dataset (2012), thus leading to have a much more complex dataset for the QA4MRE task. In any case, we consider that the graph-based multi-level linguistic representation have performed well in this particular task.

6. Conclusion

In the work reported in this paper, we have proposed a graph-based multi-level linguistic representations for texts. We have employed graph theory for formally defining a way of representing lexical, morphological, syntactical and semantic features of a text into a single graph. The graph-based representation proposed can contain words, word lemmas, PoS tags, phrasal or grammatical tags, and it can even have semantic relationships such as synonymy, hyperonymy or antonymy. The capability of containing multiple levels of natural language formal definitions in a single structure makes it a rich representation of features that allows to extract useful information as compared to other text representations reported in literature. This claim is based on the fact that other models represent the documents, considering the complete sequence of words and without taking into account that other word relationships may occur (without an implicit sequence of the words being implied).

Another contribution of this paper is the proposed MinText technique that allows to extract multi-level linguistic features by traversing minimum paths in the graph and counting these linguistic features. Although we could find other kind of paths different than the minimum one, we consider that the minimum path will contain the most representative contextual information for the words that are taken as initial and final node. We have relaxed the problem of searching the combinatorial number of minimum paths by suggesting to focus in a fixed initial node. This assumption might be valid in particular natural language problems, but it may be not so useful in other applications. The features extracted from the graph may be used in several ways, for instance, by introducing them to machine learning methods as feature vectors or to be used as representative vectors in a document collection. In any case, these vectors contain information associated to multiple linguistic levels which is a clear advantage of the proposal presented here, over other models of representation.

In order to analyze the performance of the representation proposed together with the MinText technique, we have conducted a set of experiments in a case study, namely, QA4MRE. We have observed that the representation model proposed allows us to find the correct answer for a given question between 23% and 42% of

the time. It is worth noting that this case study is highly challenging, as can be seen from the result Table 4, presented in the previous section.

This kind of result shows that the methodology may not always obtain the correct result for the QA4MRE task. The performance of this methodology may vary according to different factors, such as on the particular graph traversal algorithm chosen. For example, let us consider the following question formulated in the QA4MRE task: "What is Annie Lennox's profession?". In this case we have to decide which one of the following possible answers is the correct one:

1. Mother
2. Nurse in a hospital
3. Farmer
4. Musician
5. Dancer

The correct answer to this question should be "musician" (cosine score: 3.39), but our method selects "Nurse in a hospital" (cosine score: 4.11), as the correct answer. This incorrect prediction made by our system is mainly due to the naive selection of initial and final nodes of the graph traversal algorithm and also because of the lack of information regarding the path length. In the evaluation carried out in these experiments, we assumed that the initial node of such traversal, for all the hypotheses, should be "root"; a decision that sometimes may lead to obtaining incorrect information. For the example above, we realize that the words "nurse" and "hospital" have been taken into account twice. Hence, the features extracted for the chosen answer are more significant (greater in number) than those of the correct answer. Using the path length and the information of the candidate answers for determining the initial node (and selecting the final nodes using the question words) should be a possible way to overcome this drawback.

As future work we are planning to test the graph-based multi-level linguistic representation in other NLP tasks such as textual entailment, semantic similarity, authorship attribution etc. We are also interested in defining in detail the manner in which other semantic relationships can be integrated in the graph-based representation. We would like to evaluate the different configurations of parsing tools, in particular, for the syntactical level parsing in the task already tested. Finally, we would like to find ways of integrating other types of text tagging (such as name entity recognition) into the representation proposed here. It may be interesting to correlate and explore the applicability of the proposed representation framework to some of the past works on concept-tagging based semantic annotation (see Piryani et al., 2013) and recommendation generation (see Singh et al., 2013) for E-books.

References

Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A., 2006. Two graph-based algorithms for state-of-the-art wsd. In: Jurafsky, D., Gaussier, É. (Eds.), EMNLP. ACL, pp. 585–593.

Bejar, I., Chaffin, R., Embretson, S., 1991. Cognitive and psychometric analysis of analogical problem solving. In: Recent Research in Psychology. Springer-Verlag.

Biemann, C., 2006. 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. Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 73–80.

Catherine De Marneffe, M., Manning, C.D. 2008. Stanford typed dependencies manual.

Croft, W.B., Turtle, H.R., Lewis, D.D., 1991. The use of phrases and structured queries in information retrieval. In: Proc. of the 14th SIGIR Conference. ACM, New York, NY, USA, pp. 32–45.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1 (1), 269–271.

Dorow, B., Widdows, D., 2003. Discovering corpus-specific word senses. In: EACL. The Association for Computer Linguistics, pp. 79–82.

Keselj, V., Peng, F., Cercone, N., Thomas, C. 2003. *N*-gram-based author profiles for authorship attribution.

Marcus, M.P., Santorini, B., Marcinkiewicz, M.A., 1993. Building a large annotated corpus of english: the penn treebank. Comput. Linguist. 19 (2), 313–330.

Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M., 2006. Graph-based word clustering using a web search engine. In: Proc. of the EMNLP 2006 Conference. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 542–550.

Mauldin, M.L., 1991. 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. ACM, New York, NY, USA, pp. 347–355.

Mihalcea, R., Radev, D., 2011. Graph-based Natural Language Processing and Information Retrieval. Cambridge University Press.

Mladenic, D., Grobelnik, M. 1998. Word sequences as features in text-learning. In: In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98), pp. 145–148.

Nicolae, C., Nicolae, G., 2006. Bestcut: a graph algorithm for coreference resolution. In: Proc. of the EMNLP 2006 Conference. Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 275–283.

Peñas, A., Hovy, E.H., Forner, P., Rodrigo, Á., Sutcliffe, R.F.E., Forascu, C., Sporleder, C. 2011. Overview of QA4MRE at CLEF 2011: question answering for machine reading evaluation. In: Petras, V., Forner, P., Clough, P.D. (Eds), CLEF (Notebook Papers/Labs/Workshop).

Peñas, A., Hovy, E.H., Forner, P., Rodrigo, Á., Sutcliffe, R.F.E., Sporleder, C., Forascu, C., Benajiba, Y., Osenova, P. 2012. Overview of QA4MRE at CLEF 2012: question answering for machine reading evaluation. In: Forner, P., Karlgren, J., Womser-Hacker, C. (Eds), CLEF (Online Working Notes/Labs/Workshop).

Piryani, R., Uddin, A., Devaraj, M., Singh, V. 2013. An algorithmic formulation for extracting learning-concepts and their relatedness in ebook texts. In: Proceedings of the MIKE'2013, Lecture Notes in Artificial Intelligence, vol. 8284, pp. 524–540.

Salton, G. (Ed.), 1988. Automatic Text Processing. Addison-Wesley Longman Publishing Co., Inc.,, Boston, MA, USA.

Singh, V.K., Piryani, R., Uddin, A., Pinto, D. 2013. A content-based eresource recommender system to augment ebook-based learning. In: Proceedings of the MIWAI'2013, Lecture Notes in Computer Science, vol. 8271, 2013, pp. 257–268.

Stamatatos, E., Fakotakis, N., Kokkinakis, G. 2001. Computer-based authorship attribution without lexical measures. In: Computers and the Humanities, pp. 193–214.

Storey, V.C., 1993. Understanding semantic relationships. VLDB J. 2 (4), 455–488.

Veronis, J., 2004. Hyperlex: lexical cartography for information retrieval. Comput. Speech Lang. 18 (3), 223–252.

Zha, H., 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: SIGIR. ACM, pp. 113–120.

Zhong, S., 2005. Generative model-based document clustering: a comparative study. Knowledge Inf. Syst. 8, 374–384.