Two Approaches for QA4MRE: Information Retrieval and Graph-based knowledge representation

Helena Gómez-Adorno, David Pinto, Darnes Vilariño

Faculty of Computer Science
Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 Sur,C.P. 72570, Puebla, Mexico
{helena.gomez,dpinto,darnes}@cs.buap.mx,
http://www.cs.buap.mx/

Abstract. In this paper we present our approaches for tackling the QA4MRE 2013 main task. We have built two different methodologies, one based on information retrieval and the other one based on graph representations of the text, additionally we have built a third hybrid methodology combining both of the previous one. The first methodology uses the Lucene information retrieval engine for carrying out information extraction employing additional automated linguistic processing such as stemming, anaphora resolution and part-of-speech tagging. This approach validates the answers based on a textual entailment assessment, lexical and semantic similarity measures. In the second methodology the documents along with its hypotheses are parsed to produce a lexical, morphological and syntactic graph representation. Thereafter, we traverse different paths on the document and the hypothesis in order to find features in those graphs by counting text components (word lemmas, PoS tags, grammatical tags). As a result of this procedure, we obtain two feature vectors for each traversed path. Finally, a cosine based similarity is calculated over the feature vectors in order to select the correct hypothesis.

Keywords: Question answering system, reading comprehension, information retrieval, graph-based representation

1 Introduction

In this paper we present the experiments carried out as part of the participation in the main task of QA4MRE@CLEF 2013. The QA4MRE task is associated with the ability of a system to understand the main ideas established in a given text. The task consists of reading a document and identifying answers for a set of questions about the information that is expressed or implied in the text. The questions are written in the form of multiple choices; each question has 5 different options, and only one option is the correct answer. The detection of the correct answer is specifically designed to require various types of inference,

and the consideration of prior knowledge acquired from a collection of reference documents [1, 2]. Answering a question about a given text in an automatic way to evaluate the understanding of that text, is a very difficult task that oftenly has been tackled in the literature through some Natural Language Processing (NLP) techniques, such as Question Answering (QA). Information retrieval and QA are related, however, QA assumes that given a query, the result must be the correct answer of that question, instead of a number of references to documents that contain the answer.

The main idea behind QA4MRE task is to answer questions based on a single document. This approach is different from that of traditional QA systems, in which they have a very large corpus for searching the requested information, which implies in some cases a very different system architecture.

Since the first edition of this task in 2011, and later in the 2012, it has provided a single evaluation platform for the experimentation with new techniques and methodologies towards giving a solution to this problem. In this sense we can take the systems presented in this conference as state-of-the-art work for this research field.

The rest of the paper is organized as follows. Section 2 describes the Developed Approaches. Section 3 presents the evaluation results in the collection of documents of the QA4MRE task at CLEF 2013. Finally, Section 4 presents the conclusions obtained, so that it outlines some future work directions.

2 Developed Approaches

We have developed two different methodologies for tacking the problem for two languages, English and Spanish. The first one based on Information Retrieval techniques, and the other one is based on text representations by means of graphs. Both of them include a general Document Processing module. In the following sections each of the methodologies are discussed.

2.1 Document Processing

This module is executed for both data sets, English and Spanish. An XML parser receives as input a corpus structured in XML format which contains all the documents, along with their respective questions and multiple choice answers, as is shown in Figure 1. The XML parser extracts the documents, questions and associated answers. It stores the questions and answers identifying them according to the document to which they belong, in order to be used in the following processes. Later, the queries associated to each document are analyzed, applying a Part-Of-Speech (POS) tagger in order to identify the "question keywords" (what, where, when, who, etc.), and the result is passed to the *hypothesis generation* module. Thereafter, *hypothesis generation* module receives the set of questions with their multiple choice answers, previously processed. We construct what we means *h*ypothesis as the concatenation of the question with each of the possible

answers. This hypothesis is intended to become the input to the Information Retrieval (IR) module, i.e., the query, as well as to the graph generation module. In order to generate the hypothesis, first the "question keyword" is identified and subsequently replaced by each of the five possible answers, thereby obtaining five hypotheses for each question. For example, given the question: **Who** is the founder of the SING campaign?. And a posible answer: **Annie Lennox**. The obtained hypothesis is: **Annie Lennox** is the founder of the SING campaign.

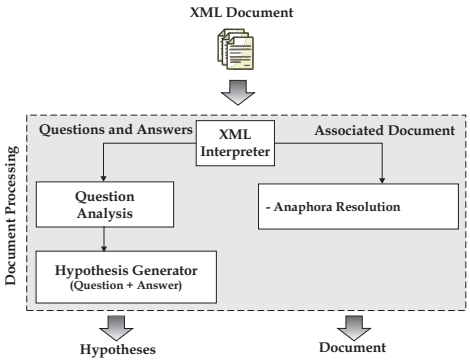

Fig. 1. Document Processing Architecture

Afterwards, we perform anaphora resolution for the English documents associated with the questions using the JavaRAP[1] system. It has been observed that applying anaphora resolution in QA systems improves the results obtained, in terms of precision [3].

The output of this module is the set of hypotheses along with its reference documents. These sets are the input for the two approaches previously mentioned.

2.2 Information Retrieval Approach

The Information Retrieval approach consists of the following two submodules: Information Retrieval (IR) and Answer Validation. Both submodules are illustrated in the Figure 2.

2.3 Information Retrieval module

The *IR module* was built using the Lucene[2] IR library. It is responsible for indexing the document collection, and for the further passage retrieval, given a

[1] http://wing.comp.nus.edu.sg/ qiu/NLPTools/JavaRAP.html
[2] http://lucene.apache.org/core/

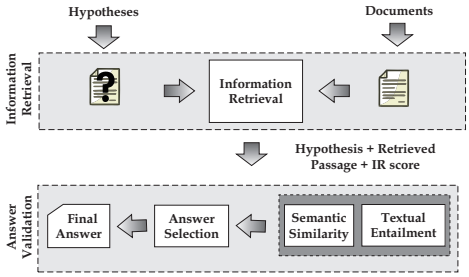

Fig. 2. Information Retrieval Approach

query. Each hypothesis obtained in the *hypothesis generation* module is processed in order to identify the query keywords, removing *stop words* (using the stop word list of python NLTK[3]). Every processed hypothesis is sent to the IR module. The IR module returns a relevant passage for each hypothesis. This passage is used as a support text to decide whether or not the hypothesis can be the right answer. For each hypothesis the first passage returned is taken (only one), which is considered the most important one. This process generates a pair "Hypothesis + Passage (H-P)", along with a lexical similarity score calculated by lucene.

Answer validation module The *answer validation* module aims to assign a score based on the textual entailment judgment to the pair H-P generated in the *Information Retrieval* module. In addition to the Textual Entailment process a semantic similarity measure is calculated over the H-P pair.

It has been proven that the textual entailment judgment may improve the performance of the hypothesis validation, given a support text, which in this case is the retrieved passage [4–6]. The aim of this module is to obtain the textual entailment judgment over all the $H - P$ pairs that it receives as input. In order to determine whether or not the passage P implies an hypothesis H, we implemented an approach based in an research work[7] presented in the Crosslingual Textual Entailment task of the SEMEVAL-2012[4]. In this work the set provided in that conference is used as a training data. The textual entailment judgment is performed over the hypotheses-passages set as test data.

For this particular problem all the previously developed models were tested, determining that the best performance is obtained when the following 29 features are used: the number of n-grams of words ($n = 1, \cdots, 4$) and characters ($n = 1, \cdots, 5$), which share each pair of sentences and the number n-grams of words ($n = 1, \cdots, 4$) and characters ($n = 1, \cdots, 5$) that are in the hypothesis and not in the support text and viceversa. In addition, the length of both sentences are included to the feature set, since it has been proven to help to obtain the textual entailment judgment. Given that this problem can be seen as a classification

[3] http://nltk.org/
[4] http://www.cs.york.ac.uk/semeval-2012/task8/

one, after several experiments, it was decided to use a 4-layer neural network, using the WEKA[5] data mining tool.

The *SemanticSimilarity* measure used in this work [8] gives a weight to each word of the sentence in terms of the degree of specificity of the word. For example the words **catastrophe** and **disaster** gain more weight than words **could** and **should**. The similarity inter-words for both sentences is integrated into this measure. The three similarity word-to-word measures proposed are Knowledge-based Measures, based on the Wordnet taxonomy (path, lin and wup)).

The similarity between the pair H y P is given by the equation 1

$$sim(H,P) = \frac{1}{2}\left(\frac{\sum_{w\ \epsilon\ \{H\}} (maxSim(w,P) * idf(w))}{\sum_{w\ \epsilon\ \{H\}} idf(w)} + \frac{\sum_{w\ \epsilon\ \{P\}} (maxSim(w,H) * idf(w))}{\sum_{w\ \epsilon\ \{P\}} idf(w)}\right) \quad (1)$$

For the *Answer Selection* process, we have developed a method based on the following rules:

1. Check the entailment judgment between the hypothesis and the recovered passage. If the judgment is "no_entailment", in the five hypotheses then this algorithm discards this answer, in other case, the lexical similarity score obtained by Lucene is used.
2. For each question, the answer obtaining the highest sum of scores is selected as the correct answer.
3. Finally, we check the Lucene score, and if the score is lower than 0.1 and higher than 0.0 we answer the question with the option "5) None of the above"; if the score is equal to 0.0 the question is not answered.

The reason for discarding the hypothesis with "no_entailment" judgment is that even thought the IR module returned a passage for the hypothesis, this one does not share sufficient information to support the selection of that hypothesis as the correct answer to the question. The use of the lexical similarity score obtained by lucene allows the system to determine which answer is more similar with its support text.

2.4 Graph-based Knowledge Extraction Approach

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 text representation in order to solve some particular problem [9]. The Graph-based Knowledge Extraction Approach consists of the following two submodules: Graph Generation and Answer Validation. Both submodules are illustrated in the Figure 3.

[5] http://www.cs.waikato.ac.nz/ml/weka/

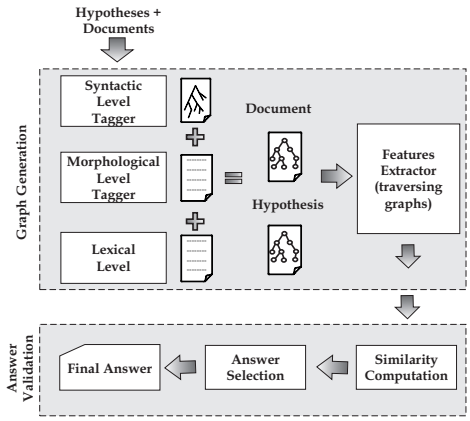

Fig. 3. Graph-based Knowledge Extraction Approach Architecture

Graph Generation module This module receives the set of hypotheses along with its associated documents from the *Document Processing* module. Text documents along with its hypotheses are parsed to produce their graph-based representation. For the graph-based representation we took into account the different linguistic levels (lexical, syntactic, morphologic and semantic) in order to capture the majority of the features presented in the text. By including those linguistic analysis we attempt to represent how different text components (words, phrases, clauses, sentences, etc) are related.

The process of the graph generation is described by the following submodules:

The *Syntactic Level Parser* is the base for the Graph-based representation. At the syntactic level we deal with rules and principles that govern the structure of a given text. Different syntactic-based parsers exist in literature, however, for the purposes of this work, we use the Stanford Dependency Parser [6] for the English language set, and Freeling[7] for the Spanish language set. In this type of parsing, we may take advantage of the grammatical relation obtained between two components of the sentence.

The *Morphologic Level Parser* deals with the identification of the morpheme's structure of a given language and others linguistic units, such as word's roots, affixes, Part-Of-Speech(POS) tags. With the aim of introducing these morphological components to the proposed representation, we have used the Stanford Log linear Part-Of-Speech Tagger[8](English) and Freeling (Spanish) in order to obtain the POS tags. Furthermore, the Lancaster stemmer algorithm was used in order to obtain truncated words.

The *Lexical Level* At this level we deal with words, one of the most basic text units, describing their meaning in relation to the physical world or

[6] http://nlp.stanford.edu/software/lex-parser.shtml
[7] http://nlp.lsi.upc.edu/freeling/
[8] http://nlp.stanford.edu/software/tagger.shtml

to abstract concepts, without reference to any sentence in which they may occur.

As a result of this process, each document is represented as a tree rooted in a $ROOT-0$ node, and 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, and a frequency label established as the number of occurrences of the pair (initial_node, final_node) in the graph plus the frequency of the dependency tag of the same pair of nodes. In the same way the hypotheses are represented as a tree with the same characteristics as well.

In Figure 4 we show the graph-based representation for the hypothesis "Annie Lennox is the founder of the SING campaign", whereas, Figure 5 shows the graph-based representation for the first sentences of the reference document associated to the given question.

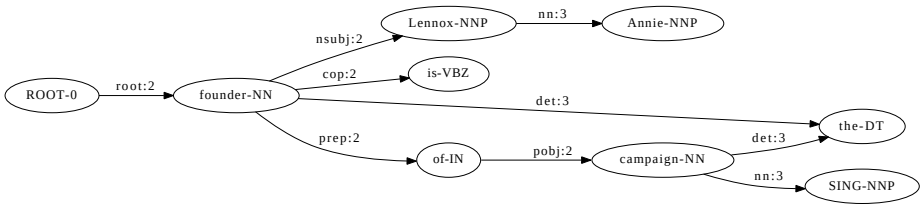

Fig. 4. Graph-based representation of the hypothesis "Annie Lennox is the founder of the SING campaign"

In the *Features Extractor module*, the process start by fixing the root node of the hypothesis graph as the initial node, whereas the final nodes selected correspond to the rest nodes of the hypothesis graph. We have used the $DijkstraAlgorithm$ [10] 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 dependencies tags found in the path. The same procedure is performed with the document graph, using as initial and final node the pair of words identified in the hypothesis. As a result of this procedure, we obtain two set of feature vectors: one for the answer hypothesis, and another one for the reference document.

Answer Validation module The answer validation module receives the set of vectorial features ($\overrightarrow{f_{t,i}}$) for each text. Thus, the reference document d will now be represented by m feature vectors ($d^* = \{\overrightarrow{f_{d,1}}, \overrightarrow{f_{d,2}}, ..., \overrightarrow{f_{d,m}}\}$), as well as the answer hypothesis h ($h^* = \{\overrightarrow{f_{h,1}}, \overrightarrow{f_{h,2}}, ..., \overrightarrow{f_{h,m}}\}$), being m, the number of different paths that may be traversed in both graphs, using the ROOT-0 vertex

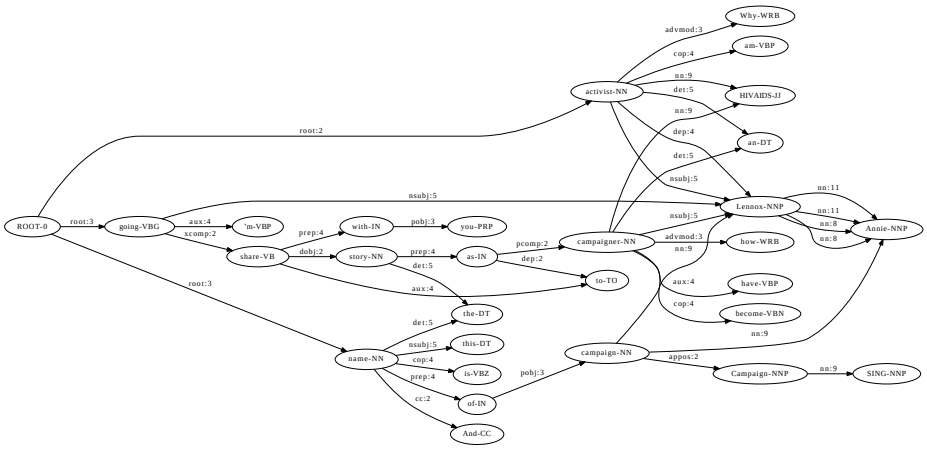

Fig. 5. Graph-based representation for one reference document

as the initial node and each one of the words appearing in the hypothesis as the final node.

Since each path of the answer hypothesis contains exactly the same number and types of components than the reference document, it is possible to calculate the degree of similarity among each path traversed. For the purposes of this study case, we have used the cosine similarity measure, which is calculated as shown in Eq. 2.

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

After obtaining all the similarity scores for the five hypothesis of one question, the hypothesis achieving the highest score is selected as the correct answer. For the experiments carried out with this methodology we have decided to answer every question. The only case in which the question is not answered is when the similarity score is 0.0, but this case only occurred in the Spanish language.

3 Experimental results

This section describes the data sets used for evaluating the methodologies proposed in this paper. Additionally, the results obtained in the experiments carried out are reported and discussed.

3.1 Corpus Description - QA4MRE task 2013

The features of the test dataset are detailed in Table 1.

It is worth to mention that this year The data set was composed of a total of 284 questions of which:

Table 1. Features of the test dataset

Features	**2013**
1. Topics	4
2. Topic details	Climate Change, Music & Society, Alzheimer and AIDS
2. Reading tests (documents)	4
3. Questions per document	15/20
4. Multiple-choice answers per question	5
5. Total of questions	240/320
6. Total of answers	1200/1600

– 240 are main questions
– 44 are auxiliary questions

The difference between main and auxiliary questions resides in the presence of a inference. In fact an auxiliary question is just a duplicate of a main question minus the inference. The idea is that the simpler versions (auxiliary) could be added to a main questions: if a system gets the difficult version wrong and the easy version right, it could be that it could not perform the required inference.

3.2 Obtained Results

The main measure used in this evaluation campaign is c@1, which is defined as shown in equation 3. This measure is defined in the QA4MRE task at CLEF 2011 with the purpose of allowing the systems to decide whether or not to answer a given question. The aim of this procedure is to reduce the amount of incorrect answers, maintaining the number of correct ones.

$$c@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n}) \tag{3}$$

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

We have sent seven runs for the English language data set and three runs for the Spanish language data set. Table 2 present the obtained results of all runs for both languages on the main questions set. The column NoA indicates the number of answered questions, the column NoU shows the number of unanswered questions, the column PcD represents the percentage of correctly discarded answers and finally the $c@1$ measure.

In particular, the runs: *buap*1301*enen*, *buap*1309*enen* were executed using the IR Approach. The difference relies in the addition of the semantic similarity measure to the *buap*1309*enen* run and all questions are answered in this run.

The inclusion of the semantic similarity measure allows to overcome the results of the *buap*1301*enen* run, but, it could achieve a better performance if we would included the "no answer" rules.

The runs: *buap*1302*enen* and *buap*1310*enen* were executed using the Graph-based Approach, with the difference of the algorithm for obtaining the shortest path between an initial and a final node. The *buap*1302*enen* run uses the *Dijkstra Algorithm*, while the *buap*1310*enen* uses *All shortest path Algorithm*, both implemented in the *Networkx*[9] tool of Python. This methodology attempts to find the similarity between the hypothesis and the complete document, without the use of IR or NLP techniques. The purpose of using this methodology was to test a framework for this particular task, but it is still a basic approach which can be improve including others pre processing techniques such as question analysis, and inferences mechanisms, and add more elements to the graph, like named entity recognition, semantic relations (synonyms, hyponyms and hyperonyms).

The runs: *buap*1303*enen* and *buap*1304*enen* are executed using a hybrid approach. It means, we mix the IR approach with the Graph-based approach. With the IR system we recovered 5 passages for each hypothesis, which we use instead if the reference document in order to find the similarity measure using the Graph-based Approach. The rest of the methodology is maintained the same. The run *buap*1304*enen* include the validation for detecting if none of the candidate answer is correct. Finally, the run *buap*1305*enen* is a voting system between *buap*1301*enen*, *buap*1302*enen* and *buap*1304*enen*. If two of the three candidates respond the same answer for a given question, that answer is selected. In other case, the question is not respond.

Table 2. Table of results Evaluation on the main questions.

Description	NoA	NoU	PcD	c@1
English Data Set				
buap1301enen	221	19	0.84	**0.27**
buap1309enen	240	0	0.00	**0.28**
buap1302enen	240	0	0.00	0.20
buap1310enen	240	0	0.00	0.19
buap1303enen	240	0	0.00	0.24
buap1304enen	240	0	0.00	**0.25**
buap1305enen	198	42	0.7	0.24
Spanish Data Set				
buap1306eses	233	7	0.86	**0.27**
buap1307eses	238	2	1.00	0.24
buap1308eses	238	2	1.00	0.23

For the Spanish language, we have sent one run *buap*1306*eses* using the Graph-based Approach and the other two runs *buap*1307*eses* and *buap*1308*eses*

[9] `http://networkx.github.io/`

using the hybrid Approach, similar to the English language runs. The best performance was obtained by the Graph-based Approach in contrary with the English results.

Table 3 present the obtained results of all runs for both languages on the auxiliary + main questions set. In this table we can observe that the results are higher than the other one. This means that our system is not able to perform the inference needed to solve the more difficult questions. The results behavior is similar to the results of the main questions data set.

Table 3. Table of results Evaluation on all questions (main + auxiliary).

Description	NoA	NoU	PcD.	c@1
English Data Set				
buap1301enen	264	20	0.80	**0.33**
buap1309enen	284	0	0.00	0.31
buap1302enen	284	0	0.00	0.24
buap1310enen	284	0	0.00	0.24
buap1303enen	284	0	0.00	0.31
buap1304enen	284	0	0.00	**0.32**
buap1305enen	240	44	0.70	**0.32**
Spanish Data Set				
buap1306eses	274	10	0.90	**0.30**
buap1307eses	282	2	1.00	0.28
buap1308enen	282	2	1.00	0.28

4 Conclusion and Future Work

We have developed two different methodologies plus a third hybrid one, as a part of our participation of the QA4MRE task 2013. The first one was built using a basic IR Approach, the second one was built by means of graph-based representations and feature extraction, finally the third hybrid approach is a combination of the two firsts approaches.

In particular we have sent ten different runs. Our best performance for the English language data set was obtained with the IR Approach, while the hybrid approach was a little bit lower. In the case of the Spanish language the Graph-based approach brings the bests results.

We consider that, even though, the Graph methodology did not achieve the bests results for the English language, it is an interesting framework to represent documents and it could be improve by adding particular characteristics of this task, such as, question analysis, question expansion (by synonym, hyponym,etc), and some improve mechanisms to allow us to detect whether to answer or not a given question.

References

1. Peñas, A., Hovy, E.H., Forner, P., Rodrigo, Á., Sutcliffe, R.F.E., Forascu, C., Sporleder, C.: Overview of qa4mre at clef 2011: Question answering for machine reading evaluation. In: CLEF (Notebook Papers/Labs/Workshop). (2011)
2. Peñas, A., Hovy, E.H., Forner, P., Rodrigo, Á., Sutcliffe, R.F.E., Sporleder, C., Forascu, C., Benajiba, Y., Osenova, P.: Overview of qa4mre at clef 2012: Question answering for machine reading evaluation. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
3. Vicedo, J.L., Ferrandez, A.: Importance of pronominal anaphora resolution in question answering systems. In: In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL. (2000) 555–562
4. Pakray, P., Bhaskar, P., Banerjee, S., Pal, B.C., Bandyopadhyay, S., Gelbukh, A.F.: A hybrid question answering system based on information retrieval and answer validation. In: CLEF (Notebook Papers/Labs/Workshop). (2011)
5. Bhaskar, P., Pakray, P., Banerjee, S., Banerjee, S., Bandyopadhyay, S., Gelbukh, A.F.: Question answering system for qa4mre@clef 2012. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
6. Clark, P., Harrison, P., Yao, X.: An entailment-based approach to the qa4mre challenge. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
7. Vilariño, D., Pinto, D., Tovar, M., León, S., Castillo, E.: Buap: Lexical and semantic similarity for cross-lingual textual entailment. In: Proceedings of the 6th International Workshop on Semantic Evaluation, Montréal, Canada, ACL (2012)
8. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: IN AAAI '06. (2006) 775–780
9. Mihalcea, R., Radev, D.: Graph-based natural language processing and information retrieval. Cambridge university press (2011)
10. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1) (1959) 269–271