Graph-based Word Clustering Applied to Question Answering and Reading Comprehension Tests

Juan Martinez-Romo and Lourdes Araujo

NLP & IR Group at UNED, ETSI Informática UNED
c/ Juan del Rosal, 16. E-28040 Madrid, Spain
{juaner, lurdes}@lsi.uned.es

Abstract. This paper describes our participation in the QA4MRE 2011 task, targeted at reading comprehension tests and multiple choice question answering. Our system constructs a co-occurrence graph with words that are common or proper nouns and verbs extracted from each document. The documents are pre-selected through an information retrieval process for recovering only those that are most relevant to a particular question. An algorithm to detect communities of words with significant co-occurrence is applied to the co-occurrence graph. Each of the detected communities are treated as different contexts of a question in the corpus, and these contexts are used to find the most suitable answer. Our evaluation results suggest that, the number of retrieved documents is an important factor in the results.

1 Introduction

The QA4MRE 2011 task has been defined as an evaluation campaign of Machine Reading systems through Question Answering and Reading Comprehension Tests. Systems are asked to return a set of answered questions by extracting knowledge from given texts and using this knowledge to find the correct answer. Questions in the test will be in the form of multiple choice questions. The task focuses on the extraction of information of single documents and the use of previously-acquired background knowledge. Our approach, as we will describe, is oriented to discover knowledge from a co-occurrence graph.

2 Assumptions

Participants are provided with questions from three topics. Associated with each topic, a reference corpus is available consisting of about 30,000 un-annotated documents related to the topic. Documents in the corpus are used to acquire the background knowledge needed to answer a test on the topic.

Each topic is composed by several tests and each test consists of one single document and a set of multiple choice questions having five options each. Participants have the option of not answering the question or to answer the question by choosing one answer in each case. Our approach makes some assumptions that will be tested in the following sections, namely:

1. The documents included in the reference collection deal with issues related the topic they are associated.

2. It is possible to find enough information in the Web to compile a background knowledge similar to that provided by organizers.

3. There may be a vocabulary gap between the questions and the answers, so the document from the test and the background can be used to bridge this gap.

4. It is better not to answer a question whose answer is uncertain than to provide a wrong answer.

5. Documents translated to different languages tend to contain redundant information. This assumption avoids the management of multilingual texts that would require additional processing time and linguistic resources.

3 System architecture

In this section we present the system developed for the QA4MRE 2011 task. Our graph-based approach is specifically designed to build a co-occurrence graph and uses the detected communities to find the correct answer to each question proposed.

3.1 Background Preprocessing

The background knowledge on which we work in this paper is a reference corpus consisting of about 30,000 un-annotated documents related to the topic.

Corpus preprocessing aims to create a structure of documents where every word is marked up as corresponding to a particular part of speech. All documents are lemmatised and PoS tagged using the GENIA tagger [1]. Instead of using all words to construct the graph, only nouns and verbs are used, since they are more discriminative than adverbs and adjectives. Accordingly, only nouns and verbs are kept and lemmatised.

3.2 Co-occurrence Graph

We consider a document to be a coherent piece of meaning, so that it is natural to make the basic assumption that all words appearing in the same document

[1] www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger

share a common meaning. Our aim is to create a link joining every two words sharing a common meaning, so co-occurrence in the same document will be taken as a proxy for this.

In order to construct the graph our algorithm extracts from each document the words that are common or proper nouns and verbs. The rest of the words are considered too common to discriminate the meaning.

The extracted words are then applied a stemming process, reducing them to their stem with the aim of increasing the significance of the number of occurrences. This is done applying the Porter algorithm [7]. After these usual preprocessing steps, the algorithm considers each pair of words (stems) and computes the corresponding link between each pair of words.

3.3 Detecting Communities

The weighted graph we have built contains the relationships between different words in the corpus. We apply an algorithm to detect communities of words with significant co-occurrence.

We have used the WalkTrap [6] program to compute communities in large networks using random walks. This software measures the similarities between vertices based on random walks.

The algorithm starts from a partition of the graph into n communities corresponding to the vertices and merges communities in order to optimize a function called modularity which measures the quality of a partition. Distances between all adjacent vertices are computed and this partition evolves by means of an iterative process. Each iteration defines a community merging which gives a hierarchical structure of communities called dendrogram. The algorithm merges the different communities taking into account the distances between adjacent vertices. The quality of the communities generated at each step is used to choose the optimal partition.

3.4 Question Answering

Each of the detected communities are treated as different contexts of a question in the corpus. In this way, the text of the question is assigned to a community and that community is considered the context of the question.

Each question is assigned to a community based on their similarity. The similarity between a question and each community is the co-occurrence of words in the text of the question and the community.

In the case of the answers the process is similar. First, each response is assigned to a community. Then, the response that has the highest similarity to the context of the question, is selected as the correct answer.

In some cases, it is not possible to assign the text of the question to just a community. Similarly, sometimes several answers get the highest similarity. In these cases the question is not answered.

4 System Scenarios

In this section we present two background collections used for the QA4MRE 2011 task. In addition, the main tuning parameters are described in the next sections.

4.1 First variant: QA4MRE-2011-EN corpus

Participants were allowed to submit a maximum of 10 runs. Also, first runs must have been produced using nothing more than the knowledge provided in the reference collection. For that reason, our first submitted runs use only the information provided in the QA4MRE-2011-EN corpus. However, additional runs could include other sources of information. Accordingly, the next section describe the other corpus used in the task.

In order to reduce the computation time, we have not used the whole reference collection in the experiments, but we have carried out a selection of the most relevant documents in the corpus to build the co-occurrence graph. We have created an index with the documents in the reference collection to retrieve and use only those that are most relevant to a particular question. The most relevant documents are obtained from the similarity measure used by the search engine implemented. The score of query q for document d correlates to the cosine-distance between the document and the query vectors in a vector space model (VSM) of information retrieval. A document whose vector is closer to the query vector in that model is scored higher.

Accordingly, we have carried out an indexing process where every document of the considered topics has been indexed by filtering stopwords extracted from a public list in the University of Glasgow[2]. For indexing tasks we used Lucene[2], which is a source information retrieval library. In addition, we have decided to analyze the impact on the results of using stemming[3]. For that, we have used the Stemming algorithm by Porter, which is available at the Snowball Web site[3].

Finally, the system has three different indices for each of the documents divided by topic. For each question, several queries are submitted to the index by retrieving a variable number of documents in each query. The number of documents retrieved in each case is a system parameter. Documents retrieved in each case are used to build the co-occurrence graph.

4.2 Second variant: IR process to build a new adapted corpus

The second variant is a slight modification of our original proposal. In this case, documents used to build the co-occurrence graph are retrieved from a commercial search engine. For that, we have used the open search web services platform of Yahoo! (BOSS[4]). Documents are obtained by submitting several queries to the

[2] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/

[3] http://snowball.tartarus.org/

[4] http://developer.yahoo.com/search/boss/

search engine composed of terms extracted from the different sources (text of the question, answers, topic). Our system performs a form of query expansion[1], a well-known method to improve the performance of information retrieval systems. In this method the expansion terms have to be very carefully selected to avoid worsening the query performance. The way in which terms are extracted and query expansion process is carried out is described in [5, 4].

4.3 Tuning parameters

The experiments generated for the task have been carried out considering different parameters in their configuration. First, we have used two reference collections; the collection provided by the organizers and another one built from a set of queries submitted to a search engine. Second, the number of documents retrieved for each query is an important factor affecting the size of the graph. Thus, several values were taken to analyze their influence on the results. Finally, in some cases the documents have been indexed by filtering stopwords and in other cases we have not used any filter.

The configuration of the different runs can be seen in Table 1.

Run	**Collection**	**# Docs**	**Filter**
uned1101enen	QA4MRE-2011-EN	100	No Filter
uned1102enen	QA4MRE-2011-EN	75	Stopwords
uned1103enen	QA4MRE-2011-EN	75	No Filter
uned1104enen	QA4MRE-2011-EN	50	Stopwords
uned1105enen	QA4MRE-2011-EN	50	No Filter
uned1106enen	QA4MRE-2011-EN	40	Stopwords
uned1107enen	QA4MRE-2011-EN	40	No Filter
uned1108enen	QA4MRE-2011-EN	20	Stopwords
uned1109enen	API Yahoo! BOSS	20	Stopwords

Table 1. System configuration and results for submitted runs. Collection: Collection used as background knowledge, # Docs: Number of documents retrieved to build the co-occurrence graph, Filter: Type of filter used to filter out words in text.

5 Results

Table 2 shows the results for the submitted runs, including answered questions, unanswered, answered right, answered wrong, unanswered right, unanswered wrong, unanswered empty, and c@1 measure.

Figure 1 illustrates the results for submitted runs per topic. Topic 3 that corresponds to Music and society obtains the best average results with respect

Run	Ans.	Unans.	A.R.	A.W.	U.R.	U.W.	U.E.	Overall c@1
uned1101enen	77	43	24	53	0	0	43	0.27
uned1102enen	63	57	17	46	0	0	57	0.21
uned1103enen	60	60	16	44	0	0	60	0.20
uned1104enen	53	67	12	41	0	0	67	0.16
uned1105enen	50	70	13	37	0	0	70	0.17
uned1106enen	45	75	11	34	0	0	75	0.15
uned1107enen	39	81	10	29	0	0	81	0.14
uned1108enen	16	104	2	14	0	0	104	0.03
uned1109enen	67	53	20	47	0	0	53	0.24

Table 2. Results for submitted runs. Ans.: Answered, Unans.: Unanswered, A.R.: Answered Right, A.W.: Answered Wrong, U.R.: Unanswered Right, U.W.: Unanswered Wrong, U.E.: Unanswered Empty, Overall c@1: c@1 measure.

to each run submitted, it even improves results of the overall. Topic 2, that corresponds to 'Climate Change', gets the best c@1 measure, and far exceeds the runs 1 and 9 the overall measure. Topic 1 (AIDS) in almost all cases has obtained a c1 measure lower than overall.

6 Conclusions

Analyzing the results, it can be said that the influence of the collection is a very important factor in the quality of the results. Although only one run (uned1109enen) has been submitted using the collection built from queries in a search engine, the difference is clear when compared with the equivalent run generated from the reference collection (uned1108enen). First, the number of answered questions is four times greater (67 vs 16) than with the collection provided by the organizers. Second, the c@1 measure is eight times higher (0.24 vs 0.03). This difference is surprising since the collection provided by the organizers should have only documents of a given topic, while queries in a search engine are performed on the whole Web.

The number of retrieved documents is also an important factor in the results. In the case of retrieving 20 documents for each query, the c@1 measure obtained is more than 9 times (0.03 vs. 0.27) than retrieving 100 documents.

Finally, it was probed that filtering stopwords has not produced a clear effect on the results, although in previous experiments that filtering provided a significant difference as to compare the results.

7 Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the project Holopedia (TIN2010-21128-C02-01) and the Re-

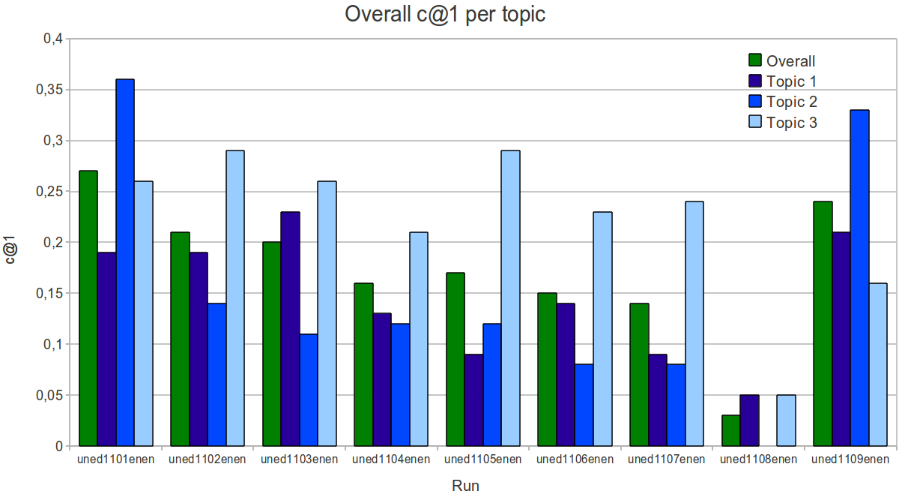

Fig. 1. Results for submitted runs per topic. Topic1: AIDS, Topic2: Climate Change, and Topic 3: Music and society.

gional Government of Madrid under the Research Network MA2VICMR (S2009/TIC-1542).

References

1. E. N. Efthimiadis. Query expansion. *Annual Review of Information Systems and Technology*, 31:121–187, 1996.
2. Otis Gospodnetic and Erik Hatcher. *Lucene in Action*. Manning, 2004.
3. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
4. Juan Martinez-Romo and Lourdes Araujo. Web spam identification through language model analysis. In *AIRWeb*, pages 21–28, 2009.
5. Juan Martinez-Romo and Lourdes Araujo. Analyzing information retrieval methods to recover broken web links. In *Advances in Information Retrieval*, volume 5993 of *Lecture Notes in Computer Science*, pages 26–37. Springer Berlin / Heidelberg, 2010.
6. Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293. Springer Berlin - Heidelberg, 2005.
7. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.