A Question Answering System for Reading Comprehension Tests

Helena Gómez-Adorno, David Pinto, and 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 it is presented a methodology for tackling the problem of question answering for reading comprehension tests. The implemented system accepts a document as input and it answers multiple choice questions about it. It 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. The proposed approach validates the answers, by comparing the text retrieved by Lucene for each question with respect to its candidate answers. For this purpose, a validation based on textual entailment is executed. We have evaluated the experiments carried out in order to verify the quality of the methodology proposed using two corpora widely used in international forums. The obtained results show that the proposed system selects the correct answer to a given question with a percentage of 33-37%, a result that overcomes the average of all the runs submitted in the QA4MRE task of the CLEF 2011 and 2012.

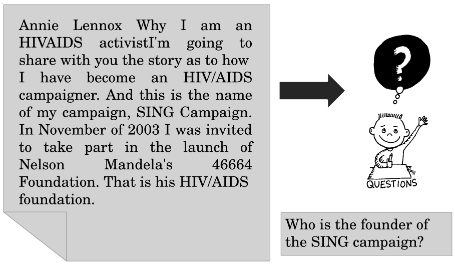
Keywords: Question answering system, reading comprehension, information retrieval, textual entailment.

1 Introduction

Reading comprehension is a task associated with the ability of a reader to understand the main ideas written in a given text. This understanding comes basically from the knowledge that is triggered to the reader by observing the different words that appear in the text. Analyzing a text is quite different than just reading it. The goal of reading comprehension is to understand the main ideas implied in the text. With the aim of evaluate the level of reading comprehension, there exist tests that ask the students to read a story or article and answer a list of questions about it. See Figure 1 in which an example of reading comprehension test is presented.

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,

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2013, LNCS 7914, pp. 354–363, 2013.



Annie Lennox Why I am an HIVAIDS activistI'm going to share with you the story as to how I have become an HIV/AIDS campaigner. And this is the name of my campaign, SING Campaign. In November of 2003 I was invited to take part in the launch of Nelson Mandela's 46664 Foundation. That is his HIV/AIDS foundation.

Who is the founder of the SING campaign?

Fig. 1. Reading Comprehension test document and question

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.

In this paper we present some experiments for exploring question answering architectures that can be applied to reading comprehension tests as an evaluation method for language understanding systems (machine reading systems). Such tests take the form of standardized multiple-choice diagnostic reading skill tests.

The main idea behind QA systems for reading comprehension tests 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.

There exist a seminal work on QA for reading comprehension tests written in the late 90's by *Hirschman* which describes an automatic reading comprehension system, **Deep Read**. The system receives as input a document, and answers questions about it. The authors designed a corpus with 60 test documents, and each document contained five associated questions, along with their correct answers. In a first approach, the system used basic techniques based on pattern matching (bag of words), enriched with automated linguistic processing (stemming, named entity identification, semantic class identification, and pronoun resolution) for extracting the sentence containing the answer. The system reported between 30-40% of precision [1].

Other studies were conducted on this research line using the same corpus. The work presented by *Charniak* [2], for instance, apply techniques that achieve small improvements over the *Hirschman's* research work. These techniques range from simple (give higher weights to verbs in the answer selection) to more complex (use specific techniques to answer specific types of questions). In the *Riloff's* work [3], a rule-based system is presented, **Quarc**. It uses heuristic rules to look for lexical and semantic clues in the question and the answer. This system finds the correct answer 40% of the time. Finally, *Hwee Tou Ng* [4] presents an approach based on machine learning techniques, in particular decision trees algorithms, with a set of features extracted from the corpus. This was the first work that achieved competitive results in the task of QA for reading comprehension tests using these techniques (39% accuracy).

The QA for reading comprehension tests field has been inactive for a long time, due to the lack of agreement in the way the systems evaluation should be done [5] . In 2011, and later in the 2012, the CLEF conference[1] proposed a QA task for *Machine Reading (MR) systems* evaluation called QA4MRE. 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 [6,7].

The QA4MRE task encourage the interest in this research line, because it provides 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 System Architecture. Section 3 presents the evaluation results in a collection of documents of the QA4MRE task at CLEF 2011 and 2012, and compares these results against those reported in literature. Finally, Section 4 presents the conclusions obtained, so that it outlines some future work directions.

2 System Architecture

The proposed architecture is made up of three main modules. Each of these modules are described in the following subsections.

2.1 Document Processing

The document processing module consists of three sub-modules: an XML Interpreter, a Query Analyzer, and a Document Pre-Processor. A detailed description of each submodule follows.

XML Parser: The XML parser receives as input a corpus structured in XML format which contains all the documents of the reading comprehension test, along with their respective questions and multiple choice answers. 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.

[1] The Cross-Lingual Evaluation Forum: `http://www.clef-initiative.eu`

Query Analyzer: This module receives as input the question set associated with the documents. A Part-Of-Speech (POS) tagger is applied to the questions in order to identify the "question keywords" (what, where, when, who, etc.), and the result is passed to the *hypothesis generation* module (this module will be explained more into detail in Section 2.2).

Document Pre-Processing: This module has the task of performing anaphora resolution for the documents associated with the questions. It has been observed that applying anaphora resolution in QA systems improves the results obtained, in terms of precision [8]. In the experiments carried out in this paper, the JavaRAP [2] system was used for anaphora resolution. It resolves third person pronouns, lexical anaphora, and identifies pleonastic pronouns.

Given that JavaRAP does not resolve anaphora of first-person pronouns, we added a process for the resolution of these cases. The process added is as follows:

1. Identify the author of the document, which is usually the first name in the document. For this purpose, the Stanford POS tagger [3] was used.
2. Each personal pronoun in the first person set PRP={"I", "me", "my", "myself"} generally refers to the author.
3. Replace each term of the document that is in the PRP set, by the document author name identified in step 1.

Take the following text for showing how this procedure works:

Emily Oster *flips our thinking on AIDS in Africa. So* ***I*** *want to talk to you today about AIDS in sub-Saharan Africa.* ***I*** *imagine you all know something about AIDS*

Step 1: ***Emily_NNP Oster_NNP*** *flips_VBZ our_PRP$ thinking_NN on_IN AIDS_NNP in_IN AfricaSo_NNP.*

In this case, the 2 first terms that have the NNP label are selected to identify the author. **Author = Emily Oster**.

Step 2: ***I_PRP*** *want_VBP to_TO talk_VB to_TO you_PRP today_NN about_IN AIDS_NNP in_IN sub-Saharan_NNP Africa_NNP ._.* ***I_PRP*** *imagine_VBP you_PRP all_DT know_VBP something_NN about_IN AIDS_NNP ._..*
In these lines two words are identify within the defined PRP set.

Step 3: *So* ***Emily Oster*** *want to talk to you today about AIDS in sub-Saharan Africa.* ***Emily Oster*** *imagine you all know something about AIDS.*

Here the words within the PRP set are replaced by the name of the author of the document.

[2] http://wing.comp.nus.edu.sg/ qiu/NLPTools/JavaRAP.html

[3] http://nlp.stanford.edu/software/tagger.shtml

2.2 Information Extraction

The information extraction module consists of the following two submodules: Hypothesis Generation and Information Retrieval. Both submodules are described as follows.

Hypothesis Generation. This module receives as input the set of questions with their multiple choice answers, which were previously processed in the module *Questions Analysis*. In this work we define hypothesis 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. 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. The process is illustrated in the following example:
Question: **Where** was Elizabeth Pisani's friend incarcerated?

Answer 1: in the Philippines
Answer 2: in the Taiwan Island
Answer 3: in the Islands of Malaysia
Answer 4: in the Greater Sunda Islands archipelago
Answer 5: in the Lesser Sunda Islands archipelago

From the previous question and their possible answers, we obtain the following hypotheses:

hypothesis 1: **in the Philippines** was Elizabeth Pisani's friend incarcerated?
hypothesis 2: **in the Taiwan Island** was Elizabeth Pisani's friend incarcerated?
hypothesis 3: **in the Islands of Malaysia** was Elizabeth Pisani's friend incarcerated?
hypothesis 4: **in the Greater Sunda Islands archipelago** was Elizabeth Pisani's friend incarcerated?
hypothesis 5: **in the Lesser Sunda Islands archipelago** was Elizabeth Pisani's friend incarcerated?

The benefit of using these hypotheses as queries for the IR module is to search passages containing words that are in both, the question and the multiple-choice answer, instead of search passages containing words from the question and the answer, independently.

Information Retrieval. The IR module was built using the Lucene[4] IR library. It is responsible for indexing the document collection, and for the further passage retrieval, given a 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[5]). Every processed hypothesis is sent to the IR module.

[4] `http://lucene.apache.org/core/`
[5] `http://nltk.org/`

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.

2.3 Answer Validation

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.

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 [9,10,11]. 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[12] presented in the Crosslingual Textual Entailment task of the SEMEVAL-2012[6]. 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 10 features are used: the number of n-grams of words and characters ($n = 1, \cdots, 5$), which share each pair of sentences. 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. Since this problem can be seen as a classification one, after several experiments, it was decided to use a 4-layer neural network, using the WEKA[7] data mining tool.

2.4 Answer Selection

For this last phase of the system, the method shown in Algorithm 1 is developed based on the following rules:

1. Check the entailment judgment between the hypothesis and the recovered passage. If the judgment is "no_entailment", then this algorithm discards this answer, in other case, the lexical similarity score obtained by lucene and the prediction percentage given by the textual entailment judgement are added.
2. For each question, the answer obtaining the highest sum of scores is selected as the correct answer.

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

[6] http://www.cs.york.ac.uk/semeval-2012/task8/

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

Algorithm 1. *AnswerSelection*

```
Input: Hypotesis
Input: Support Text
Input: $lucene\_score$ : Lexical similarity score given by lucene
Input: $te\_prediction$: Textual entailment Prediction percentage given by weka
Input: $te\_judgment$ : Textual entailment judgment
Output: List of correct answer for each question
foreach pair $(a_k = Hypotesis_k + Support\_text_k)$ $d_i\_q_j$
where $i = 1 \ldots 12, j = 1 \ldots 10, k = 1 \ldots 5$ do
    if $judgment =$ "$no\_entailment$" then
        discards that possible answer;
    else
        $score[d_i, q_j, a_k] = lucene\_score + te\_prediction$;
foreach $i, j, k$ in score do
    if $mayor[i, j] < score[d_i, q_j, a_k]$ then
        $mayor[i, j] = score[d_i, q_j, a_k]$;
        $mayorId[i, j] = k$;
foreach $i, j$ in $d_i\_q_j$ do
    return i, j, mayorId[i,j]
```

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. The textual entailment prediction value obtained through the Weka tool adds extra information when the correct answer is selected.

3 Experimental Results

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

3.1 Corpus Description - QA4MRE Task

In order to determine the performance of the system proposed in this paper we used the corpora provided in the QA4MRE task of the CLEF 2011 and 2012. The features of the two test datasets are detailed in Table 1.

3.2 Obtained Results and Error Analysis

The main measure used in this evaluation campaign is c@1, which is defined as shown in equation 1. 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

Table 1. Features of the two test datasets (QA4MRE 2011 and 2012 tasks)

Features	**2011**	**2012**
1. Topics	3	4
2. Topic details	Climate Change, Music & Society, and AIDS	Climate Change, Music & Society, Alzheimer and AIDS
2. Reading tests (documents)	4	4
3. Questions per document	10	10
4. Multiple-choice answers per question	5	5
5. Total of questions	120	160
6. Total of answers	600	800

a given question. The aim of this procedure is to reduce the amount of incorrect answers, maintaining the number of correct ones.

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

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

Table 2 presents the obtained results. It can be observed that in both data sets, the 2011 data set and the 2012 data set, the average over all best runs and over all runs were exceeded.

Table 2. Comparison of the results obtained by our QARCT system and the Average Scores over all runs and over best runs

Description	**2011**	**2012**
QARCT	**0.33**	**0.37**
Avg. over all best runs	0.28	0.32
Avg. over all runs	0.21	0.26
Random baseline	0.20	0.20

For evaluation purposes of this task and given the evaluation measure, it is considered that is better do not give answer at all than provide an incorrect answer. In that sense, we analized the number of questions that have been incorrectly answered. According to this error analysis, it was considered important to experiment with other passages division models: using n sentences rather than a single one, defining a window of n words that should include the passage. Unfortunately, this experiment did not improve the precision of the system, therefore it was discarded.

After an extensive review of the passages retrieved by each hypothesis, it was noticed that the *IR module* is not consistent returning, in the first place, the passage containing the answer to the question, but instead the passages were returned sometimes in the second, third or even in the fifth place. This issue makes the validation step useless for these cases. It is therefore necessary that the *IR module* retrieves a 100% of the correct passages. As a possible solution it is planned to extend the number of the recovered passages from 1 to 5, for each hypothesis.

Another problem found in the *Answer Selection* module is that the lexical similarity score given by lucene is not enough to capture the similarity between the hypothesis and the support text, when they do not share the same words. To overcome this problem, two things can be done: 1) To include a query expansion module trying to add synonyms, hyperonyms, etc, in order to obtain a higher lexical similarity, and 2) To add a semantic similarity algorithm which can discover the degree of similarity between two sentences, even though they do not share the same words exactly. For example in the hypothesis: "she esteems him is Annie Lennox 's opinion about Nelson Mandela", the recovered passage is "Everyone one in the world respects Nelson Mandela, everyone reveres Nelson Mandela"; but the score assigned by lucene is too small and it does not select that answer as the correct one. The addition of semantic similarity score will help to raise the score of this two phrases and select the correct answer because it will probably find the relation between the words "esteems", "revers" and "respect".

4 Conclusion and Future Work

In this paper we have presented a complete methodology for tackling the problem of question answering for reading comprehension tests. Additional modules can be added to this methodology, or maybe a refinement of each step presented may be done. However, we consider that the proposal is complete in terms of such modules needed in order to solve the aforementioned problem.

The implementation of the first person anaphora resolution algorithm helped Lucene to find more precise information for retrieving more accurately those passages that contain the possible answer. This type of anaphora resolution was not implemented in the original software used in the experiments, therefore, we consider this contribution very important in this research work.

By adding the textual entailment module to the basic measures based on lexical similarity, it allowed to correctly answer a higher number of questions. Additionally, this module allowed to determine whether or not to answer the given question, which have a high impact in the final scores of the proposed system.

We have compared the performance of the system presented in this paper with those reported in the QA4MRE task of CLEF 2011 and 2012. We observed that the obtained results overcome the average of the runs submitted to that conference. There still is more research to do as future work.

It is planned to analyze the use of Machine Learning techniques for the answer validation module. For this purpose, it is necessary to determine the features that fulfills an answer when it is correct or incorrect. Based on these features, a classifier should be trained in order to obtain a model capable of identify whether or not an answer is correct or not. Additionally, we are considering to implement semantic similarity measures with the aim of improving the level of matching between the hypothesis and the possible passages of the target text, when these two sentences do not share the exactly same words, but those that are semantic similar.

References

1. Hirschman, L., Light, M., Breck, E., Burger, J.D.: Deep read: a reading comprehension system. In: Proceedings of the 37th meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA (1999)
2. Charniak, E., Altun, Y., de Salvo Braz, R., Garrett, B., Kosmala, M., Moscovich, T., Pang, L., Pyo, C., Sun, Y., Wy, W., Yang, Z., Zeller, S., Zorn, L.: Reading comprehension programs in a statistical-language-processing class. In: ANLP/NAACL Workshop on (2000)
3. Riloff, E., Thelen, M.: A rule-based question answering system for reading comprehension tests. In: Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Sytems. ACL, Stroudsburg (2000)
4. Ng, H.T., Teo, L.H., Lai, J., Kwan, J.L.P.: A machine learning approach to answering questions for reading comprehension tests. In: Proceedings of EMNLP/VLC (2000)
5. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. Nat. Lang. Eng. 7(4), 275–300 (2001)
6. 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)
7. 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)
8. Vicedo, J.L., Ferrandez, A.: Importance of pronominal anaphora resolution in question answering systems. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 555–562. ACL (2000)
9. 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)
10. 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)
11. Clark, P., Harrison, P., Yao, X.: An entailment-based approach to the qa4mre challenge. In: CLEF (Online Working Notes/Labs/Workshop) (2012)
12. 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. ACL, Montréal (2012)