Solving Open-Domain Multiple Choice Questions with Textual Entailment and Text Similarity Measures

Neil Dhruva[‡], Oliver Ferschke[†‡] and Iryna Gurevych[†‡]

† Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for International Educational Research

‡ Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science
Technische Universität Darmstadt

{dhruva|ferschke|gurevych}@ukp.informatik.tu-darmstadt.de

Abstract In this paper, we present a system for automatically answering open-domain, multiple choice reading comprehension questions about short English narrative texts. The system is based on state-of-the-art text similarity measures, textual entailment metrics and coreference resolution and does not make use of any additional domain specific background knowledge. Each answer option is scored with a combination of all evaluation metrics and ranked according to their overall score in order to determine the most likely correct answer. Our best configuration achieved the second highest score across all competing system in the entrance exam grading challenge with a c@1 score of 0.375.

1 Introduction

Question Answering (QA) systems have always been a key focus of Information Retrieval and Natural Language Processing research. Such systems aim to automatically answer questions posed in natural language. The objective of the Entrance Exams task in the CLEF Question Answering Track is the creation of a system for determining the correct answers to a set of multiple choice reading comprehension questions about English narrative texts.

The task demands a deep understanding of a short passage of text and is closely related to the QA4MRE[1] main task [18], which aims to focus on the concept of answer validation. However, the main difference between the main task, and the Entrance Exams task is the fact that no additional knowledge (background material) is provided for the latter. The aim is thus to evaluate automatic answering systems under the same conditions as humans are evaluated, considering that humans have certain background knowledge from real-life experiences. A detailed task definition is provided in Section 2.

Based on conclusions drawn from a state-of-the-art analysis, which is discussed in Section 3, our system uses three main components: coreference resolution, text similarity and textual entailment. Text-Hypothesis (T-H) pairs are generated for each entrance

[1] Question Answering for Machine Reading Evaluation

exam using combinations of sentences from the text, questions about the text and the corresponding answer options. These T-H pairs form the basic processing units for similarity and entailment analyses, which are employed to determine the correct answer to a question. We describe our approach and the architecture of our system in Section 4.

Finally, we present the results obtained with different system configuration in Section 5 and provide a detailed error analysis in Section 6. We close with a summarization of our findings and a discussion of future research directions in Section 7.

2 Task Definition

The objective of the task tackled in this paper is to identify the correct answer option for a multiple choice question in a reading comprehension test for a given English narrative text. Since the questions are not restricted to a particular topic, they require a wide range of inference. Additionally, no background information is provided and systems are supposed to determine answers based on common sense knowledge that high school students are supposed to possess.

The dataset provided in this task is composed of reading comprehension tests taken from Japanese university entrance exams and is available in English, Russian, French, Spanish and Italian. The labeled training data consists of 12 documents with 60 questions and an average of 5 questions for each document. The test data comprises 12 documents with a total of 56 questions. We consider a *test item* to be a combination of the narrative text, a single question about this text and all answer options for this question. An *exam* is then the set of all test items with the same text. Each exam is evaluated with a score between 0 and 1 using the c@1 measure [16], which is defined as:

$$\text{c@1} = \frac{1}{n}\left(n_c + n_u \frac{n_c}{n}\right)$$

where, n_c is the number of correctly answered questions, n_u, the number of unanswered questions, and n is the total number of questions. The main idea of this metric is to encourage systems to reduce the number of incorrect answers while maintaining the number of correct ones by leaving some questions unanswered.

3 Related Work

The question answering task on university entrance exams was first introduced in this form at the CLEF QA4MRE Lab in 2013. In this section, we review the previous work from the pilot task. Related work on enabling technologies such as textual entailment is introduced later in the paper.

The best performing system in the 2013 entrance exam task was presented by Banerjee et al. [3]. Their answer determination module comprises of named entity recognition, syntactic similarity measures as well as textual entailment. They select the correct answer using a maximum weighted score algorithm, but more importantly, they have successfully developed a technique for avoiding questions for which the system shows low confidence rather than answering them wrongly. Overall, the system achieved a $c@1$ score of 0.42 by answering 13 of the 23 attempted questions correctly.

The runner up system in the performance ranking was presented by Li et al. [12] with an overall $c@1$ score of 0.35. The system heavily relies on coreference resolution in the narrative text. Moreover, it includes a sentence extractor to identify the most relevant sentences to a particular question while the final answer is determined with a textual entailment classifier. However, the authors conclude that their sentence extractor was the main bottleneck and that the textual entailment component used in their system did not perform up to the mark.

Finally, Arthur et al. [2] employ the same system [1] they proposed for the QA4MRE 2013 main task [18]. It makes no use of textual entailment and fails to clear the baseline ($c@1 = 0.25$) with a $c@1$ score of 0.22. The authors conclude that the reliance on statistical methods alone, and the absence of logical inference hinders the answer determination ability of their system for this task.

As discussed by Peñas et al. [15], using purely statistical analyses of words to determine the correct answer to a reading comprehension questions does not work well for this task. Using textual entailment, on the other hand, provides a means to carry out necessary inferences to determine the correct answer. Banerjee et al. [3], who created the best performing system in the pilot task, also highlight the use of text similarity measures in addition to textual entailment. With these conclusions at hand, we present our system based on textual entailment, text similarity and coreference resolution.

4 System Architecture

We now present the architecture of our question answering system which is based on text similarity measures, textual entailment and coreference resolution. The general idea of the system is to retrieve sentences from the text that are relevant for a given question and to identify the correct answer option among the multiple choices based on textual entailment and similarity between each option and the retrieved sentences.

Our system builds upon the Darmstadt Knowledge Processing Software Repository (DKPro) [10], which is based on the Apache UIMA Framework [6]. In particular, it uses NLP components from DKPro Core[2] and textual similarity measures from the DKPro Similarity[3] [4] framework.

The system is divided into six main modules: corpus reader, preprocessing, sentence retrieval, answer similarity analysis, entailment analysis, and answer selection. Figure 1 gives a schematic overview of the system architecture. The individual components are described in the remainder of this section.

4.1 Corpus Reader

The corpus reader parses the provided XML document with the entrance exam data and initializes a separate UIMA Common Analysis Structure (CAS) for each test item. The text of each CAS consists of the narrative text appended with the question and the corresponding answer options. Both the question and the answer options are then marked-up

[2] http://dkpro-core-asl.googlecode.com

[3] http://dkpro-similarity-asl.googlecode.com

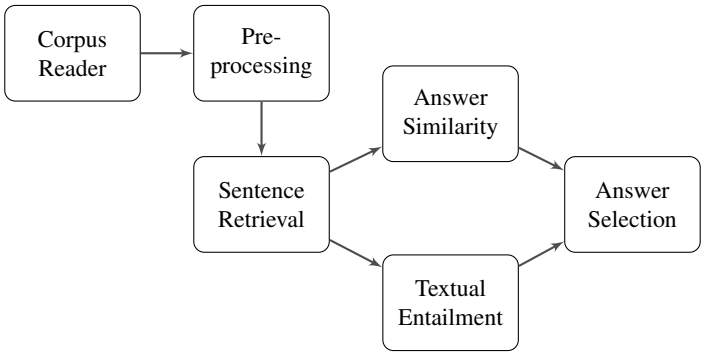

Figure 1. High-level overview of the system architecture

with UIMA annotations which identify the exact span of text and hold additional meta information contained in the XML source document. The final CAS is then passed on to the next module in the pipeline for further processing.

4.2 Preprocessing

The preprocessing module consists of several NLP components that annotate the text before the similarity and entailment analysis is carried out. All of these components are based on the Stanford CoreNLP[4] package and employed in DKPro Core wrappers. We use the Stanford segmenter and lemmatizer for identifying tokens, sentences and lemmas in the text. The Stanford parser [11] is furthermore used to annotate parts of speech (PoS) and the constituent structure. Moreover, we observed that most of the narrative texts feature recurring characters (as also noted by Li et al. [12]). Hence, the Stanford Named Entity Recognizer [7] is used to identify entities of the *Person* type in the text. Finally, the Stanford coreference resolver [17] links the different named entities with other occurrences in the text. Based on the resulting coreference chains, an answer option is then modified, or *resolved*, such that pronouns in the answer option are replaced with the corresponding proper nouns obtained from the question text. For example:

Question: *What did John want?*
Answer Option: *He wanted a bike.*
Resolved Answer: *John wanted a bike.*

To further improve the sentence retrieval, we also resolve all pronoun entities in the main text. This resolution facilitates the identification of relevant sentences in the text for each question and answer option, since it reduces the lexical variation introduced by referring expressions.

[4] http://nlp.stanford.edu/software/corenlp.shtml

4.3 Sentence Retrieval

We found that the downstream textual entailment module is sensitive to irrelevant information. That is, the longer the text from which a given hypothesis is to be inferred, the more likely the system is to report a falsely positive inference. It is therefore necessary to first identify sentences in the text that are most relevant to a particular *question* before proceeding with further analyses of the answer options. We use three different text similarity measures to identify sentences from the document that are most relevant to a particular question:

1. **Lexical similarity**: The Word n-gram Jaccard measure from the DKPro Similarity package implements a generalization of the Ferret measure [13] to support similarity calculations for token n-grams of arbitrary length using the Jaccard coefficient. We employ the Jaccard measure based on token unigrams.
2. **ESA-based similarity**: Text similarity based on explicit semantic analysis [8] using a Wikipedia based index. For the similarity calculation, cosine is used as an inner product along with L2 vector normalization in the cector comparator provided by DKPro Similarity.
3. **PoS similarity**: A part of speech based measure for calculating the overlap between two bags of PoS. We only consider nouns, verbs and adjectives for the similarity calculations.

For all similarity measures, the input texts are stopword-filtered using a subset of the English stopword list from the DKPro WSD[5] package. Each similarity measure provides a score between 0 and 1. Using a linear combination of these scores, we select the top k sentences for each question. In our experiments, we empirically found $k = 5$ to be an optimal value for our setup. We use only these sentences in the following modules to determine the correct answer among all possible options.

As a basic processing unit for the following analyses, we define *Text-Hypothesis (T-H) pairs*. We use each retrieved sentence from the narrative text as T and all answer options along with all possible combinations of the question with each answer option as H. Both for T and H, we use resolved and unresolved version of the text, depending on the system configuration.

We noticed that, for many questions, the correct answer lies in the sentence before or after the selected sentence. Hence, instead of restricting the set of T entities entirely to the selected sentences, we added an option to include the previous and next sentences as well. These are added as separate T entities to the set. This setting is referred to as $S^{\pm}$ later on (see Table 1).

All downstream analyses are then performed on all possible T-H pairs. Our approach to T-H pair creation is prone to overgeneration, i.e. it will contain irrelevant pairs that can even degrade the system performance. We therefore implemented several selection techniques that only consider particular T-H pairs. An overview of the different configurations will be described in Section 5.

[5] http://dkpro-wsd.googlecode.com

4.4 Textual Entailment

In order to compute textual entailment on the T-H pairs, we use the EXCITEMENT Open Platform (EOP) [14], a UIMA-based framework with implementations of state-of-the-art textual entailment algorithms along with pre-trained models. Each T-H pair is processed using an EOP annotator, which first carries out additional preprocessing on the T-H pair (depending on the model used) and then uses an Entailment Decision Algorithm (EDA) to perform the inference between T and H.

We use the `MaxEntClassificationEDA` configuration provided in EOP, which uses a maximum entropy model based on the OpenNLP MaxEnt [5] package for learning an entailment classifier. Scoring techniques, such as bag of words and bag of lemmas scoring, are used along with an option to employ additional resources including Verb Ocean and WordNet. This enables us to incorporate real-world knowledge to aid the entailment procedure. The classifier was trained on the RTE-3 dataset [9]. The entailment decisions are binary but provide a confidence score. We store both the decision and the confidence score in the T-H markup of the CAS.

4.5 Answer Similarity

While the idea of T-H pairs originally stems from the textual entailment theory, we generalize the idea to similarity analysis as well. As mentioned previously, each sentence (or resolved sentence) is labeled as text (T), while an answer option (or resolved answer option, or a question combined with an answer option) is labeled as the hypothesis. Consequently, similarity measures can be applied to these T-H pairs and corresponding scores can be used to determine the correct answer.

We explained earlier that we do not only include the answer options (and their resolved variants) in the set of hypotheses, but also a combination of the answer option with the question. The reason for this can be seen in the following example:

Question: *People are normally regarded as old when*
Answer Option: *they are judged to be old by the society*
Combined: *People are normally regarded as old when they are judged to be old by the society.*
Relevant Sentence: *But in general, people are old when society considers them to be old, that is, when they retire from work at around the age of sixty or sixty-five.*

It is easy to see that the combination of question and answer makes the inference of the correct answer easier as compared to using the answer option alone.

We use the same similarity measures that we described earlier in Section 4.3 to measure the similarity between T and H in all T-H pairs. A linear combination of the scores generated by each measure is stored in the markup of each T-H pair and used together with the entailment score for the final answer selection.

4.6 Answer Selection

The entailment confidence score and the answer similarity scores obtained for each T-H pair are finally used to select the correct answer option. We calculate a *correctness score*

Table 1. Overview of individual system configurations and their performance. A=Answer, Q=Question, S=Selected sentence in text, $S^{\pm}$=Selected sentence, previous sentence and next sentence. The prefix *re* indicates the resolved version of an item, e.g. reA indicates a resolved Answer. + indicates concatenation, comma separated series indicate alternatives from which the highest score is chosen.

#	Sentence Selection		Answer Similarity			Entailment		Performance		
	Sentence	Measures	Text	Hyp.	Measures	Text	Hyp.	Corr.	Incorr.	c@1
1	Original	Jaccard	–	–	–	S	A, reA	11	45	.196
2	Original	Jaccard, ESA	$S^{\pm}$	A, reA	Jaccard, ESA	$S^{\pm}$[a]	A, reA, Q+A	11	45	.196
3	Original	Jaccard, ESA	–	–	–	$S^{\pm}$	A, reA, Q+A	16	40	.286
4	Original	PoS, ESA	$S^{\pm}$	A, reA	ESA	$S^{\pm}$	A, reA, Q+A	13	43	.232
5	Original	Jaccard	$S^{\pm}$	A, reA	Jaccard, ESA	$S^{\pm}$	A, reA, Q+A	16	40	.286
6	Original	ESA	$S^{\pm}$	A	Jaccard, ESA	$S^{\pm}$	A, Q+A	14	42	.250
7	Resolved	Jaccard, ESA	re$S^{\pm}$	reA	Jaccard, ESA	re$S^{\pm}$	reA, Q+A	**21**	**35**	**.375**

[a]Sentences are selected based on the top 10 sentences according to their similarity to each answer option rather than the sentence selection procedure described in the text.

by using a linear combination of the two scores as mentioned below:

$$\text{correctness score} = x(\text{entailment score}) + y(\text{similarity score})$$

We experimented with different linear combinations of the two scores and found that the best performance is achieved when the entailment score is given a higher weight compared to the similarity score. More specifically, we achieved good results with $\{x, y\} = \{2, 1\}$ and $\{x, y\} = \{3, 2\}$. In cases where the entailment confidence exceeded a threshold of .90, we set the weights to $\{x, y\} = \{1, 0\}$ thus eliminating the similarity score from the decision process. We finally select the answer option corresponding to the T-H pair with the highest correctness score as the correct answer for a given question.

5 Evaluation

For the Entrance Exams 2014 task, we submitted a total 7 configurations. In this section, we discuss the individual configurations, the scores achieved with each setup and finally provide a detailed error analysis.

Each configuration consists of three main components, the sentence selection strategy, the answer similarity analysis and the entailment analysis. The exact setup for each configuration is provided in Table 1. Overall, configuration 7 achieved the best results with a c@1 score of 0.375. Moreover, configurations $3, 5$, and 6 performed above or equal to the random baseline of 0.25, while $1, 2$, and 4 performed poorly with c@1 scores below the baseline. The reason configuration 1 failed was mainly due to the fact that answer selection was merely performed on the sentences pertaining to the questions, without considering those surrounding the selected sentences. Hence the idea of using the sentence before and after a selected sentence, was not implemented for this configuration.

Configuration 2, on the other hand, used the previous and next sentences, but failed to clear the baseline. One of the main reasons for this was the fact that sentences chosen

as H for the entailment analysis were chosen based on high answer similarity scores rather than the sentence selection procedure based on the question. This resulted in the selection of certain sentences that were related to incorrect answer options, and not directly related to the question. For configurations 4, the linear combination of answer similarity and entailment scores used to determine the final answer was ineffective, and led to incorrect answers. Configurations 3 and 5 did well because the weight assigned to the entailment score was increased, and a lower weight, (0 in case of configuration 3,) was assigned to the answer similarity score. Nonetheless, answer similarity helped determine a few answers where entailment analysis alone had failed, for instance, in configurations 5 and 7.

Additionally, in all but the last run, we used only the original sentences as T. However, for 7, we introduced the idea of using resolved sentences as T. This improved our scores dramatically, giving a c@1 score of 0.375. As for configuration 6, without the use of resolved answers, it failed to perform as well as $3, 5$ or 7.

This leads us to conclude that coreference resolution in the sentences and answer options is highly beneficial in determining the correct answer. Additionally, using the previous as well as the next sentence in addition to the selected sentence as T is better than using only the selected sentence. The results confirm that many answers can be identified from a sentence close to the selected one using the question text. Finally, when using a linear combination of the answer similarity and entailment scores, a higher weight should be assigned to the entailment score.

6 Error Analysis

In order to identify systematic errors in the decision making process of our system, we conducted an error analysis on the output of each configuration described in the previous section.

The implementation of coreference resolution replaces pronouns and phrases describing a person entity with the proper noun. This generates some issues when the phrase describing a person is vital to answer determination. The coreference resolution in the preprocessing module can further be improved by replacing only certain terms, mainly pronouns, in sentences with the corresponding named entities.

While the sentence extraction works well in most cases, there are two problems that can be encountered. The first problem is that the correct sentence pertaining to a question is not the one with the highest score. Consequently, we select k (=5) sentences rather than just one. Nonetheless, an improvement in identifying the most relevant sentence, and reducing k to 1 or 2 sentences will help to narrow down possible mistakes with answer determination. The second problem is when the question is not sufficient to identify the correct sentence in the text. For example, a question like the following requires sentence selection based on the answer options instead of the question text:

Question: *The main point the author wishes to make is that*

While the answer similarity module was able to assist in determining certain answers, it was generally outperformed by the other measures employed in our system, in particular by the entailment module. It particularly underperformed in case of similarly phrased

answer options for a given question, which caused false predictions. Consequently, a lower weight was assigned to the similarity score. With an improved linear combination of the various similarity measures integrated in this module, the performance can further be improved.

While the entailment module based on EOP provides good results, especially in the case of resolved sentences and answer options, it does not perform optimally in certain scenarios. For instance, the module does not perform well when more than one answer options are very similar. This is illustrated by the following example:

Excerpt: *"I'm from Georgia," he said in a loud voice,"and proud of it." "Sorry I made the mistake," I told him, though I just couldn't see what difference it made whether he came from Georgia or Alabama.*
Question: *What mistake did the writer make about the man with glasses?*
Incorrect Answer: *He thought the man came from Georgia*
Correct Sentence: *He thought the man's home state was Alabama*

Moreover, certain questions require advanced inference based on world knowledge rather than relying on the document surface text alone. In these cases, the correct answers are difficult to determine with the entailment module. For example:

Excerpt: *He was holding onto the arms of his seat so tightly that the blood had left his fingers. He was what is known as a "white-knuckle flier." He would not look out the window, and he was sweating so much that the stewardesses had to keep bringing him towels.*
Question: *What is meant by a "white-knuckle flier?"*
Answer: *It is a person who is extremely nervous on an airplane.*

We also encountered errors when a question and answer pair spanned multiple sentences. Since our system considers at most 3 sentences at a time, i.e. the selected, previous and next sentence, it sometimes fails to obtain the desired inference for the correct answer option. With a more generalized retrieval module that supports larger text windows, we could be able to circumvent this problem but have to take into account the added noise that is introduced by larger spans of text.

Thus, to summarize, a refined approach to coreference resolution along with an improved answer similarity module will help to improve scores to a great extent. In addition, the entailment module can be trained with different datasets to improve the classifier, while a tighter sentence selection module will help to narrow down possible sentences relevant to a particular question.

7 Conclusions and Future Work

Automatically answering multiple choice reading comprehension questions is a challenging task. In this paper, we presented a system designed to solve such questions without any domain-specific background knowledge. Our strategy for determining answers to reading comprehension questions is based on three main components: text similarity, textual entailment and coreference resolution.

As already proposed in related work, textual entailment is the key technology for tackling this task. However, it is sensitive to irrelevant information both in the text and the hypothesis, so a strong filter or retrieval component based on text similarity measures drastically helps to improve the quality of the entailment results. Due to the nature of this task, which focused on reading comprehension questions about narrative texts, coreference resolution furthermore helped to improve the discriminative power of the entailment and similarity scores for the final answer selection.

With our best configuration, we achieved a $c@1$ score of 0.375, which puts our system on the second place in the overall performance ranking. Moreover, 3 other configurations performed better than or equal to the random baseline of $c@1 = 0.25$. In our error analysis, we finally identified the most prominent error types our system encountered.

As part of future work, our priorities are to refine the coreference resolution module, and to work on improving the performance of the answer similarity module. In addition, the entailment module can be trained on datasets more suited to the task, in order to improve the classifier, while a tighter sentence selection module will help narrow down possible sentences relevant to a question.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806.

References

1. Arthur, P., Neubig, G., Sakti, S., Toda, T., Nakamura, S.: Inter-Sentence Features and Thresholded Minimum Error Rate Training: NAIST at CLEF 2013 QA4MRE. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes. pp. 1–11 (2013)
2. Arthur, P., Neubig, G., Sakti, S., Toda, T., Nakamura, S.: NAIST at the CLEF 2013 QA4MRE Pilot Task. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes. pp. 2–5 (2013)
3. Banerjee, S., Bhaskar, P.: Multiple Choice Question (MCQ) Answering System for Entrance Examination. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes (2013)
4. Bär, D., Zesch, T., Gurevych, I.: DKPro Similarity : An Open Source Framework for Text Similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 121–126 (2013)
5. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. Computational Linguistics 22(1), 39–71 (1996)
6. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10(3-4), 327–348 (2004)
7. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370. Morristown, NJ, USA (2005)

8. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of The Twentieth International Joint Conference for Artificial Intelligence. pp. 1606–1611. Hyderabad, India (2007)
9. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third pascal recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 1–9. RTE '07 (2007)
10. Gurevych, I., Mühlhäuser, M., Müller, C., Steimle, J., Weimer, M., Zesch, T.: Darmstadt knowledge processing repository based on uima. In: Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology. Tübingen, Germany (2007)
11. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. vol. 1, pp. 423–430. Morristown, NJ, USA (2003)
12. Li, X., Ran, T., Nguyen, N.: Question Answering System for Entrance Exams in QA4MRE. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes (2013)
13. Lyon, C., Barrett, R., Malcolm, J.: A theoretical basis to the automated detection of copying between texts, and its practical implementation in the ferret plagiarism and collusion detector. Plagiarism: Prevention, Practice and Policies (2004)
14. Padó, S., Noh, T.G., Stern, A., Wang, R., Zanoli, R.: Design and realization of a modular architecture for textual entailment. Natural Language Engineering pp. 1–34 (2013)
15. Peñas, A., Miyao, Y., Hovy, E.: Overview of QA4MRE 2013 Entrance Exams Task. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes. pp. 2–7 (2013)
16. Peñas, A., Rodrigo, A.: A simple measure to assess non-response. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 1415–1424 (2011)
17. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 492–501 (2010)
18. Sutcliffe, R., Peñas, A., Hovy, E., Forner, P.: Overview of QA4MRE Main Task at CLEF 2013. In: Proceedings of CLEF 2013 Evaluation Labs and Workshop. Online Working Notes (2013)