Testing lexical approaches in QA4MRE

Hugo Rodrigues, Luísa Coheur, Ana Cristina Mendes,
Ricardo Ribeiro, and David Martins de Matos

Spoken Language Systems Laboratory - L^{2}F/INESC-ID
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 – 1000-029 Lisboa, Portugal
{hugo.rodrigues, luisa.coheur, ana.mendes, ricardo.ribeiro,
david.matos}@l2f.inesc-id.pt

Abstract. In this paper we describe our strategy in the course of our participation in the 2012 QA4MRE main task. We follow a lexical approach, based on both Word Proximity and similarity measures. In the former, we implement a method that was successfully applied in the "Who Wants to be a Millionaire" contest; in the later we use the notion of "extent", that is, a passage that includes terms of the given questions or answers, and results from comparing the attained extents through widely known similarity measures such as Jaccard and Dice. Considering the 2011 QA4MRE competition, our results are promising, although still far from the ones attained by the winning system.

Keywords: Machine Reading, Word Proximity, Distance Measures, Similarity Measures, QA4MRE

1 Introduction

Machine Reading (MR) aims at developing systems capable of "reading" and extracting knowledge from free text. However, in the same way computers do not play chess as humans do, systems do not interpret text as we do. We are able, for example, to quickly understand to which entity a pronoun refers, which is still not true for machines: in the sentences *The friends sat on the chairs because they were tired* and *The friends sat on the chairs because they were cozy*, the pronoun *they* refers, in the former, to the friends and in the second sentence to the chairs. We can disambiguate with no effort, but machines cannot.

In order to boost MR, a task dedicated to this topic – Question Answering for Machine Reading Evaluation (QA4MRE) – was introduced in the Cross-Language Evaluation Forum (CLEF), in 2011 [10]. Being given a text and several questions about it, competing systems had to choose the correct answer among five candidates to each question, showing in this way their level of "comprehension" of the text. The information needed to correctly choose between the different questions could be found in the given texts and in a collection of documents, called Background Collection.

Our main motivation to participate in this task is related with the FalaComigo project, where an agent poses multiple-choice tests to the audience. At the current moment these tests are manually crafted, although we have already implemented a system

capable of generating questions [4] and distractors. Our approach to the QA4MRE tasks represents our efforts in developing a tool that selects an answer from a set of possible candidates, as our goal is to automatize all the multiple-choice tests generation.

In this paper we investigate a lexical approach to the QA4MRE task. We study the contribution of an algorithm previously applied to the "Who Wants to be a Millionaire?" contest, as its goal is also to choose the correct answer among several possible answers, and the usage of similarity measures to assess the likeness of the questions and answers.

This paper is organized as follows: in Section 2 we present related work, in Section 3 we detail our approaches and, in Section 4, we evaluate and discuss them. In Section 5 we present the main conclusions and point to some future work.

2 Related Work

Twelve systems participated in the QA4MRE task in 2011; however, only eight working notes are available [10]. For all submitted runs (43 for english, 11 for german and 9 for romanian), nearly half reported a score below the baseline, which was of 20% accuracy (considering that each question has 5 different answers and that there is an uniform distribution of the different answers, the baseline is attained by always choosing the n-th answer). The winning system [9] achieved results of 0.57 considering the c@1 measure, as defined in Section 4 [10].

Although very different approaches were followed by participating systems, several steps were common to many of them, and various resources were widely exploited. In fact, many systems performed pre-processing, namely: anaphora or co-reference resolution [2, 14], stopword filtering [13] and Named Entity (NE) Recognition (NER) [5]. Regarding tools and resources, Lucene[1] was used by many systems to index the texts (as described by Iftene et al. [5] and Martinez-Romo and Araujo [7]) and WordNet [8] was a constant presence (for instance Saias and Quaresma [13] report its use in synonym detection).

Nevertheless, different strategies were implemented, from information retrieval- to logic-based approaches. For instance, Verberne [14] took advantage of the BM25 function to rank passages from the Background Collection according to their similarity to a given text fragment, "expanding" it in this way. In the work described by Iftene et al. [5], Lucene is used to index the texts. The built index is then queried using the questions, creating this way another index, built with the retrieved passages/documents. Then, based on this new index, answers are used as queries in a new retrieval step. The relevance scores from each retrieval step are then used to compute a final score for each answer. Babych et al. [1], on the other hand, describes a system for German based in logical inferences. Here, given the candidate answer C, the input text T, and the Background Collection B, the system tries to infer if $(T \wedge B) \vdash C$. Text is parsed in a dependency graph, and hyponym and other relations are extracted from this graph.

Other strategies try to relate terms. Saias and Quaresma [13] report the use of rules to measure the distance between the key elements of the question and the answer. The system described by Cao et al. [2] tries to simulate the strategy applied by people when

[1] http://lucene.apache.org/

learning a new language and answering reading tests. According to these authors, people will first locate named entities in the passages related with the questions. Thus, their system performs NER to find related passages and, afterwards, compares the NEs between the question and the passages. Terms are also related by using WordNet relations, such as synonym and hypernym. Each type of relation has a weight associated, which contributes to the final score.

A completely different approach is reported by Martinez-Romo and Araujo [7]: the system links all nouns (proper and common) and verbs within a given document, establishing a co-ocurrence graph. This means that the terms appearing in a given document are related under the same topic. Then, WalkTrap [12] is used to automatically discover "communities", that is, clusters that gather terms belonging to the same topic. Then, each question is assigned to a community based on their similarity. Following this, each answer is also assigned to a community; the selected answer is the one with greater similarity to the question context (i.e., community). We should note that the authors do not specify what are the similarity measures used to compare questions with communities, answers with communities, and the communities themselves.

In what concerns the winning system [9], it combines two different strategies: an Answer Validation (AV) approach and a Question Answering (QA) approach. The best results were accomplished by using the system as an hybrid between the two. The AV module is based on textual entailments: for each answer of a given question, an hypothesis H is generated, according to a set of patterns. These are then used to retrieve passages from the texts, which are indexed with Lucene. The topmost sentence, T, is paired with the corresponding hypothesis, resulting in the pair T-H. These pairs are then processed by a pipeline of different strategies to check if they are textual entailments. Among these strategies are the comparison of NEs, the number of co-occurring unigrams, bigrams and skip-bigrams between T and H, and the matching of question and answer types. Finally, the pair with greatest score from all strategies is chosen as correct answer. In what concerns the QA module, it starts by doing a similar task. Following some rules, each question is transformed into a pattern, where the wh-word is substituted by one of the candidate answers. From these patterns are also extracted stopwords, creating a keyword list. Then, each pattern will be compared against each sentence from the documents. If they do not match, the same is done between the respective keywords list and the sentences. Whichever matches, a score is assigned. Finally, the answer associated with the sentence with greater score is chosen as the correct answer.

3 Lexical approaches

Considering our participation in the 2012 QA4MRE task, we detach two of the submitted runs. The first employs `Word Proximity`, a strategy based on previous work to solve the "Who Wants to be a Millionaire?" contest [6]. The second is based in the same strategy, but also uses similarity measures to compare passages related to the question with passages related with the answer and use those measures to evaluate their similarity.

3.1 Word Proximity

`Word Proximity` technique is based on the assumption that answers occur close to questions terms. Originally, the algorithm was applied to documents retrieved from the Web. In the present work, we will apply it to each reading test text. The algorithm calculates the distance between each candidate answers' term and the question terms in the surroundings. It weighs the distances, of a maximum radius[2], so that documents with too many references to an answer but not to the corresponding question terms worth less. The algorithm is presented in Algorithm 1. The parameter *documentSplit* represents an array where each position is a term in the document.

Algorithm 1 Pseudocode for word proximity scoring algorithm, giving more weight to answers near question words, within radius words.

```
DistanceScore(documentSplit, questWords, ansWords, radius)
score, ansFoundWords = 0
for i = 1 to ||documentSplit|| do
    if documentSplited[i] ∈ ansWords then
        ansFountWords += 1
        for j = (i−radius) to (i+radius) do
            if documentSplited[j] ∈ questWords then
                score += (radius −|i − j|)/radius
            end if
        end for
    end if
end for
if ansFoundWords == 0 then
    return 0
else
    return score/ansFoundWords
end if
```

The best value for *radius* is not trivial to obtain, but according to the authors it is about 40-50 [6, Figure 1]. Both answers and questions can be filtered from stopwords, from wh-words (*who*, *how*) to prepositions (*a*, *from*) or 'to be' forms (*are*, *was*).

3.2 Similarity Measures

Our second approach is based on similarity measures. For this we use the notion of extents, that is, a passage that includes each term of a given query at least once [3]. The used queries are simply the questions and answers, seen as bag of words. Thus, we will have an extent for the question (question extent) and other five extents, one for each answer (answer extents). As the original constraint is too strong (all terms in the query must appear in the extent), we created a different version of the concept. This is based

[2] We use the term radius as it was introduced by Lam et al. [6], instead of window. Notice that the size of a window is twice the radius.

on Part of Speech (POS) tagging. The idea is to have important words (read nouns and verbs present in the query) to contribute with some weight to the extent. The extent has a score threshold, which, if surpassed, defines the extent. Table 1 shows the scores attributed to each POS. The threshold is defined by two parameters: the tag threshold and the *others* threshold. The later is set, empirically, to 8.0, while the former is defined in function of the query, and is set to half the total present in the query. This way we can create extents that contain only parts of the query (thus, reducing their size), but that are still large enough to apply the similarity measures (for example, if the tag threshold is set to 12.0, and we find three Proper Nouns together (3 times 4.0), we still need to find other eight words (8 times 1.0) to complete the extent[3]). This strategy is called `Extents Points`.

POS Tag	Score
Proper Noun	4.0
Common Noun	2.5
Verb	1.5
Others	1.0

Table 1: Scores for each POS tag.

The extents are then compared against each other (question extent versus each one of the answer extents) and choose as correct answer the one with highest similarity value. The most widely used similarity measures that do not penalize word order are used in our experiments: `Overlap`, `Jaccard` and `Dice`, as defined in Equations 1 to 3.

$$\text{Overlap}(X,Y) = \frac{|X \cap Y|}{\min(|X|,|Y|)} \tag{1}$$

$$\text{Jaccard}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2}$$

$$\text{Dice}(X,Y) = 2 \times \frac{|X \cap Y|}{|X| + |Y|} \tag{3}$$

4 Evaluation

In this section we detail the experimental setup and we show the achieved results with the two most relevant submitted runs.

4.1 Experimental Setup

Considering the QA4MRE evaluation in 2011, the given texts – TED talk transcriptions – dealt with three topics: “Aids”, “Climate Change” and “Music and Society”. In 2012

[3] Whenever necessary, this expansion is done evenly for both sides of the extent.

a new topic was considered: "Alzheimer". Each topic has four reading tests associated, with a text and ten questions each. Thus, the exercise comprises a total of 160 questions. Each question has five hypothesis of answer, from which only one was correct. The corpus characteristics can be consulted in Table 2. The other source of knowledge, the Background Collection, was not used in our experiments.

Track	Number questions	Total words	Unique words	Longest question	Shortest question	Avg. Length
2012	160	1762	672	23	3	11

Table 2: Characteristics of 2012 QA4MRE corpus.

The evaluation in QA4MRE is done with c@1 [10], which can be defined as:

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

where n_R is the number of correct answers and n_U the number of unanswered questions, among n questions. The metric rewards systems that choose not to answer questions instead of doing it and getting it wrong. Note that c@1 ends up being accuracy if we answer all questions.

Our system only does not answer questions to which all candidate answers get no score. Also, if a question is negative, that is, contains a *not*, then the answer with the least score is chosen. The idea behind this option is that the given candidate answer is less related with the question and, thus, it is the less probable answer to the question due to the presence of the *not* [6].

4.2 Results

As stated before, we detach in this paper two of the submitted runs (Run 3 and Run 4). These two runs were based in the evaluation done with the 2011 corpus and were the ones that attained the best results. The first is only based on Word Proximity, with a radius of 20 (value defined after some experiments with the 2011 corpus). The other run combines both described techniques. Here, `Word Proximity` used, once again, a value of 20 for the radius, and `Extents Points` was ran with `Dice` as similarity measure. Results are shown in Table 3.

4.3 Discussion

The results accomplished are promising (approximately 0.34 in c@1, in both runs), if a comparison is made with the 2011 results. Although the corpus is fairly different (different tests, texts and questions), most of the participating systems had results bellow 0.25 in c@1. However, our results are still far away from the 0.57 attained by Pakray et al. [9].

Run 3				Run 4			
Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**
0.41	0.38	0.29	0.26	0.44	0.35	0.32	0.28
0.33				0.34			

Table 3: Results by topic for the two submitted runs.

The evaluation by topic shows that our system does not perform equally in different domains. This may have to do with a non purposeful difficulty increase (that is, tests are harder but were not meant to) or simply with the topic domain (i.e., more technical and, thus, requiring more synonyms knowledge). If we go deeper in the evaluation (by test), we see that this problem is even more patent, with scores ranging from 0.10 to 0.55 within the same topic.

Considering our techniques, previous results showed that `Word Proximity` performs better when using smaller values for radius (20). We also noticed that, with the current algorithm, a question term closer to the answer has more weight than two or three question terms in the extremes of the considered snippet. A problem arises: which one is better, proximity or quantity? The answer can be found by developing other algorithms.

In what respects similarity measures it was clear, on previous experiments, that `Overlap` distance is not accurate for this task, performing as good as the baseline, much because it will boost extents containing other extents. This is due to the fact that the ratio between the size of the intersection (and, because one extent encloses the other, this is the smallest extent) and the size of the smallest extent, following `Overlap` definition, ends up being 1.0 for all those candidate answers.

It is also important to note that we used no other resources, namely the Background Collection. We believe, thus, that results can be better when using such information.

5 Conclusions and Future Work

In this paper, we explored the application of a lexical approach to QA4MRE, a Machine Reading task. In particular, we used the `Word Proximity` algorithm [6], which had been previously employed in the "Who Wants to be a Millionaire?" contest. We also tested a set of similarity measures between passages of text containing terms of the question and passages containing the answer candidates.

Regarding future work, much can be done. Both approaches can be boosted, either by testing with other measures, or by pre-processing the texts (for instance, by using a stemmer or WordNet for synonyms). The use of the Background Collection, that was not considered in our experiments, may improve our results, as well as other combinations of the developed techniques.

Acknowledgments

This work was supported by national funds through FCT Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2011. Thanks are also due to QREN (Quadro de Referência Estratégica Nacional), Fundo Europeu de Desenvolvimento Regional (EU) and AdI (Agência de Inovação) for financial support to FalaComigo project (QREN number 13449).

References

[1] Svitlana Babych, Alexander Henn, Jan Pawellek, and Sebastian Padó. Dependency-based answer validation for german. In Petras et al. [11].

[2] Ling Cao, Xipeng Qiu, and Xuanjing Huang. Question answering for machine reading with lexical chain. In Petras et al. [11].

[3] Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 358–365, New York, NY, USA, 2001. ACM.

[4] Sérgio Curto, Ana Cristina Mendes, and Luisa Coheur. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue and Discourse*, 3(2): 147–175, March 2012.

[5] Adrian Iftene, Alexandru-Lucian Gînsca, Mihai Alex Moruz, Diana Trandabat, and Maria Husarciuc. Question answering for machine reading evaluation on romanian and english languages. In Petras et al. [11].

[6] S.K. Lam, D.M. Pennock, D. Cosley, and S Lawrence. 1 billion pages = 1 million dollars? mining the web to play "who wants to be a millionaire?". In *Uncertainty in Artificial Intelligence (UAI2003)*, pages 337–345, Acapulco, Mexico, 2003. URL http://www.grouplens.org/papers/pdf/1m-uai2003.pdf.

[7] Juan Martinez-Romo and Lourdes Araujo. Graph-based word clustering applied to question answering and reading comprehension tests. In Petras et al. [11].

[8] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38: 39–41, November 1995. ISSN 0001-0782.

[9] Partha Pakray, Pinaki Bhaskar, Somnath Banerjee, Bidhan Chandra Pal, Sivaji Bandyopadhyay, and Alexander F. Gelbukh. A hybrid question answering system based on information retrieval and answer validation. In Petras et al. [11].

[10] Anselmo Peñas, Eduard H. Hovy, Pamela Forner, Álvaro Rodrigo, Richard F. E. Sutcliffe, Corina Forascu, and Caroline Sporleder. Overview of qa4mre at clef 2011: Question answering for machine reading evaluation. In Petras et al. [11].

[11] Vivien Petras, Pamela Forner, and Paul D. Clough, editors. *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, 2011.

[12] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In Pinar Yolum, Tunga Güngör, Fikret S. Gürgen, and Can C. Özturan, editors, *ISCIS*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293. Springer, 2005. ISBN 3-540-29414-7.

[13] José Saias and Paulo Quaresma. The di@ue's participation in qa4mre: from qa to multiple choice challenge. In Petras et al. [11].
[14] Suzan Verberne. Retrieval-based question answering for machine reading evaluation. In Petras et al. [11].