An Approach to the Main Task of QA4MRE-2013

Marília Santos, José Saias and Paulo Quaresma

Departamento de Informática, ECT
Universidade de Évora, Portugal
m9210@alunos.uevora.pt, {jsaias,pq}@uevora.pt

Abstract. This article describes the participation of a group from the University of Évora in the CLEF2013 QA4MRE main task. Our system has a superficial text analysis based approach. The methodology starts with the preprocessing of background collection documents, whose texts are lemmatized and then indexed. Named entities and numerical expressions are sought in questions and their candidate answers. Then the lemmatizer is applied and stop words are removed. Answer patterns are formed for each question+answer pair, with a search query for document retrieval. Original search terms are expanded with synonyms and hyperonyms. Finally, the texts retrieved for each candidate response are segmented and scored for answer selection. Considering only the main questions, the system best result was obtained in the third run, having answered to 206 questions, with 0.24 c@1 and 51 correct answers. When evaluating main and auxiliary questions, the final run continued to have our better results, being answered 245 questions, with 64 right answers and 0.26 for c@1. The use of hypernyms proved to be an improvement factor in the third run, which results had a 12% increase of correct answers and a 0.02 gain in c@1.

Keywords: MRE, QA, NLP

1 Introduction

This article describes the participation of a group from the University of Évora in the Question Answering for Machine Reading Evaluation (QA4MRE) challenge of the 2013 edition of Cross Language Evaluation Forum (CLEF)[1]. Although some authors of this paper have previous work in other QA4MRE editions [4,5], this work is based on a new system for the QA4MRE Main Task, associated with the first author's master's thesis work, and focused on the English language.
The objective of this task is the automatic understanding of one or more texts, and the subsequent identification of the answer for several questions about information that is stated or implied in those texts. While answering the questions, systems must process single documents, and Background Collections (BC) with documents that can be used as auxiliary information sources [2].

[1] http://clef2013.org/

This year's QA4MRE Main Task was composed by 4 topics, namely "Aids", "Climate Change", "Music and Society" and "Alzheimer", and all of them having a background collection of documents. Each topic had 4 reading tests with 15 to 20 questions each, and each question had 5 choice answers [1]. The test was composed by 240 main questions and 44 auxiliary questions. The latter are duplicates of the main questions, but without the previously required inference, allowing to test the ability of systems to use inference and its impact in the question treatment.
Next section presents our system arquitecture. Section 3 describes the methodology we used to process the questions, answers and the background information. The evaluation of the obtained results is detailed in section 4, while the last two sections are devoted to an analysis of those results, some conclusions and a balance of our participation.

2 Architecture

The system architecture is shown in Figure 1 and has the following components:

• XML Parser - Extracts texts, questions and answers from the input and stores them on the system;

• Indexing Component - Documents from BC pass through the lemmatizer (Candc tools/ C&C Boxer[2]) and then they are indexed with Lucene [3];

• Consult Index Component - Responsible for processing question and answers and perform document retrieval. With keywords from question and answers, this component uses Lucene to search for relevant documents in BC. The analysis and search query creation is based on:

- Lemmatizer - Question and answers's words are parsed to the corresponding lemma form;
- Named Entity Recognition (NER) - Through regular expression, the system tries identify entity names or mentions;
- WordNet module from Natural Language Toolkit[4]: the system uses synonyms, derivationally related forms and hypernyms;
- Numerical expressions - Through regular expression, the system tries identify numerical expressions;
- Remove stop words.

• Filter Component - Responsible for select relevant text segments, assigning a score to each segment and to each candidate answer. This component applies a set of criteria to choose the most plausible answer.

[2] http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer
[3] Apache Lucene is an open source information retrieval software library. http://lucene.apache.org/
[4] http://nltk.org

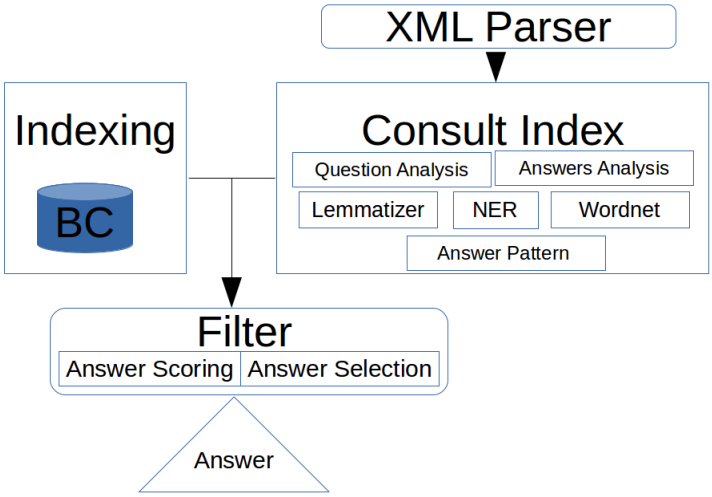

Fig. 1. System Architecture

3 Methodology

The system is based on a simple approach without a deep linguistic processing. In this edition of QA4MRE, our system generated 3 runs, having minor differences in configuration, as explained below. The processing performed on the BC texts, the reading tests and questions, comprises the following steps:

1. Indexing Component (this component is used only once)
   (a) Lemmatization is applied to the text of all documents in BC;
   (b) BC documents are indexed, considering the lemmatizer outcome;
2. XML Parser
   (a) The information from the input is extracted and stored on the system;
3. Consult Index Component - Each question is processed with the following steps and as illustrated in the examples:
   (a) Entities and numerical expressions from question and candidate answers are stored in the system. The filter uses them to score answers and text segments;

```
How can Alzheimer's patients regain the sense of smell?
1 through chemotherapy
2 through clinical trials
3 through treatment with bexarotene
4 by lying in the sun
5 None of the above

Entities: Alzheimer's patients
```

(b) Question and candidate answers pass through the lemmatizer;

```
How can Alzheimer's patient regain the sense of smell?
1 through chemotherapy
2 through clinical trial
3 through treatment with bexarotene
4 by lie in the sun
5 None of the above
```

(c) Stop words are removed from question and candidate answers;

```
Alzheimer's patient regain sense smell
1 chemotherapy
2 clinical trial
3 treatment bexarotene
4 lie sun
5 none
```

(d) For each pair (question, candidate answer) try to form an Answer Pattern. The Answer Pattern is compound by: keywords from question; keywords from answer; synonyms, derivationally related forms and hypernyms (used only on the third run) from each keyword;

```
- Alzheimer's
synonyms: Alzheimer's disease | hypernyms: dementia
- patient
hypernyms: case
- regain
synonyms: recover | related forms: recoverer | hypernyms: get
- sense
hypernyms: awareness
- smell
hypernyms:  sensation
- chemotherapy
related forms: chemotherapeutical | hypernyms: therapy
- clinical
related forms: clinic
- trial
synonyms: test | hypernyms: attempt
- treatment
related forms: treat | hypernyms: care
```

(e) Document retrieval, using Lucene to get relevant documents, using the generated Answer Patterns to querying over the indexed BC;

```
Query:
((Alzheimer's OR dementia OR Alzheimer's_disease) OR (case
OR patient) OR (regain OR recoverer OR recover OR get) OR
(awareness OR sense) OR (smell OR sensation)) OR
((chemotherapy OR chemotherapeutical OR therapy)) OR
```

```
((clinic OR clinical) OR (test OR trial OR attempt)) OR
((care OR treatment OR treat) OR (bexarotene)) OR
((lie) OR (sun))
```

4. Filter Component - For each question:
   (a) Each document is validated for each Answer Pattern:
      – If it doesn't contain 50% keywords from question and 50% keywords from answer, it is discarted;
      – If the answer has a numerical expression which does not exist on the document, it is discarted;
      – If the answer or the question has entities and if the document does not contain 30% of them, it is discarted;
   (b) When a document is valid:
      – Each Answer Pattern that validates the current document receives a score with the sum of:
         • Number of entities in the text;
         • Number of numerical expressions in the text;
         • Number of times that each keyword, from current Answer Pattern, occurs in the text;
      – The document score is the sum of each of its Answer Patterns score;
   (c) Thereafter, a second analysis is performed, only on the top 5 resulting documents from the filter; (This step is used only in the first and in the third runs)
      – Documents are split into text segments;
      – Current Answer Pattern's score is incremented if 80% of Answer Pattern's words are present in the current text segment and the distance between them is less or equal to 5;
   (d) Answer Selection:
      – If the filter returns no relevant documents, then the system selects the answer "*5 - None of the above*";
      – The system returns *Unanswer* when there is more than one maximum, or in cases where there is a small difference between the maximum and another answer's score;
      – If none of above applies, the system returns the answer with maximum score.

The difference between the runs is reflected in the number of answers given, and in system's accuracy. This difference can be observed in the following examples:

Example 1: `How can Alzheimer's patients regain the sense of smell`?
Unanswered in the first and the second run;
Answered correctly in the third run.

Example 2: `How can apolipoprotein E help people with Alzheimer's`?
Answered wrongly in the first and the second run;
Answered correctly in the third run.

Example 3: `What is U.S. AIDS policy dominated by`?
Unanswered in the second run;
Answered correctly in the first and the third run.

Examples 1 and 2 are cases where the use of hypernyms causes a small improvement on Component Filter. Example 3 shows the importance of applying the methodology step 4.c when the information is not dispersed.

4 Results

In QA4MRE, the evaluation of all runs submitted is based on the c@1 measure, discussed in [3]:

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

Equation (1):
n_R - number of correctly answered questions;
n_U - number of unanswered questions;
n - total number of questions.

4.1 Evaluation on the main questions

In the first approach the system answered to 188 of 240 questions, of which only 45 were correct, resulting in 0.23 c@1. In the second run, 185 questions were answered, with 0.18 c@1. And in the last run we answered to 206 questions, with 0.24 c@1 and 51 correct answers. Table 1 shows the detail of the system result assessment, by topic and by run.

4.2 Evaluation on all questions

For the first run, the system answered to 224 out of 284 questions. From those, 57 were correctly answered, and the c@1 was 0.24. In the second, 219 questions were answared. The c@1 was 0.19. In the final run, our system answered to 245 questions, finding 64 right answers and obtaining 0.26 for c@1. Table 2 shows these results with greater detail.

5 Discussion

One of the main causes of this system failure is the lack of an entities disambiguation module, because entities are, quite often, referred by different expressions. Other identified causes are:

Table 1. Results of the main questions

	Topic	Unanswered	Answered Right	Answered Wrong	c@1
Run 1	Alzheimer	19	11	30	0.24
	Music and society	15	12	33	0.25
	Climate Change	9	14	37	0.27
	Aids	9	8	43	0.15
	Total	52	45	143	0.23
Run 2	Alzheimer	18	9	33	0.19
	Music and society	18	11	31	0.24
	Climate Change	9	10	41	0.19
	Aids	10	5	45	0.10
	Total	55	35	150	0.18
Run 3	Alzheimer	14	14	32	0.29
	Music and society	9	15	36	0.29
	Climate Change	5	14	41	0.25
	Aids	6	8	46	0.15
	Total	34	51	155	0.24

1. Yes/no questions;
2. Answers supported by adverbs of frequency (rarely, always, never, sometimes, ...);
3. Words with high frequency have a negative impact in our system due to way the scoring algorithm works. This is specially noticed when it causes the selection of non relevant documents and incorrect answers and, in this way, it invalidates the possibility of answering "*5 - None of the above*". These failures were observed essencially for the Aids topic.

We have also observed that using a second analysis in the Filter Component (step 4.c in the methodology section) is only effective when the information about the correct answer is not disperse over several documents. However, the use of this approach allowed the improvement of 5-8% relatively to the base option (run 2), with the exception of the topic "Music and Society", where there was no impact. The use of hyperonyms didn't cause any improvement in the Aids topic but in the "Alzheimer" and "Climate Change" topics it allowed an improvement of 10% relatively to the base option and in the "Music and Society" topic an improvement of 5%.

6 Conclusion

We described the experience in QA4MRE challenge, using a simple system, with a superficial text analysis based approach. This system clearly needs further developments, aiming to improve the analysis of the questions and answers.

Table 2. Results of the main + auxiliary questions

	Topic	Unanswered	Answered Right	Answered Wrong	c@1
Run 1	Alzheimer	19	11	30	0.24
	Music and society	21	16	41	0.26
	Climate Change	10	17	47	0.26
	Aids	10	13	49	0.21
	Total	60	57	167	0.24
Run 2	Alzheimer	18	9	33	0.19
	Music and society	26	15	37	0.26
	Climate Change	10	13	51	0.20
	Aids	11	7	54	0.11
	Total	65	44	175	0.19
Run 3	Alzheimer	14	14	32	0.29
	Music and society	12	20	46	0.30
	Climate Change	6	18	50	0.26
	Aids	7	12	53	0.18
	Total	39	64	181	0.26

Namely, we intend to work on the disambiguation of entities, establishment of relations between acronyms and entities, and trying to handle the failure causes described in the previous sections. One of the critical aspects is to change the way our system evaluates answer patterns composed by words with high frequency; we need to add a new component to improve the answer selection process and, namely, to take into account the question and answer types. We have also detected that the incorporation of an anaphora resolution module would allow the system to answer more questions and to improve its performance.
On a more abstract level, we intend to assess the strengths of the system used by Évora's team last year and combine strategies with some new ideas tested in this year's work.

References

1. QA4MRE 2013, `http://celct.fbk.eu/QA4MRE`
2. QA4MRE@CLEF2013. Track Guidelines, `http://celct.fbk.eu/QA4MRE`
3. 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 - Volume 1. pp. 1415–1424. HLT '11, Association for Computational Linguistics (2011)
4. Saias, J., Quaresma, P.: The di@ue's participation in qa4mre: from qa to multiple choice challenge. In: Petras, V., Forner, P., Clough, P.D. (eds.) CLEF 2011 Labs and Workshop: Notebook Papers. Amsterdam, The Netherlands (2011)

5. Saias, J., Quaresma, P.: Di@ue in clef2012: question answering approach to the multiple choice qa4mre challenge. In: Proceedings of CLEF 2012 Evaluation Labs and Workshop - Working Notes Papers. Rome, Italy (September 2012)