Implementation of the ArabiQA Question Answering System's Components

Yassine Benajiba, Paolo Rosso, Abdelouahid Lyhyaoui

Abstract—Most of the components of a Question Answering (QA) system are language-dependent. For this reason, it is really necessary to build such a system component by component, taking into consideration each of the target language peculiarities. In our case, we are interested in building an Arabic QA system (ArabiQA). In previous works, we have presented how we managed to implement an Arabic-oriented Passage Retrieval (PR) system and a Named Entities Recognition (NER) system for Arabic texts, two of the necessary components to build ArabiQA. In this work we present the different components that ArabiQA will contain making a special focus on the Answer Extraction (AE) module. This module is required to achieve high precision, because it is responsible for extracting the different possible answers from the relevant passages retrieved by the PR module and it represents the last step before validating the extracted answers and presenting them to the user. The obtained results show that this approach can help significantly to tackle the AE task.

Index Terms—Answer Extraction, Arabic Natural Language Processing, Question Answering.

I. INTRODUCTION: BACKGROUND AND MOTIVATIONS

Information Retrieval (IR) systems were thought and designed to return relevant documents related to the user’s query. These systems had a great success because they allowed the internet users to find the needed documents in a fast and efficient way. However, IR systems are unable to satisfy a special kind of users who are interested in obtaining a simple answer to a specific question (with both the question and the answer being formulated in natural language). The research work of the first author was supported partially by MAEC - AECl. We would like to thank the PCI-AECI A7067/06 and MCyT TIN2006-15265-C06-04 research projects for partially funding this work.

Y. Benajiba is a Ph.D. Student (AECI grant holder) at the Dept. of Informatic Systems and Computation of the Polytechnic University of Valencia, Spain. e-mail: ybenajiba@dsc.upv.es

P. Rosso, Ph. D. is a member of the Natural Language Engineering Laboratory and a permanent lecturer in the Dept. of Informatic Systems and Computation of the Polytechnic University of Valencia, Spain. e-mail: proso@dsc.upv.es

A. Lyhyaoui, Ph. D. is a member of the Engineering Systems Laboratory and a permanent lecturer in the Abdelmalek Essaadi University, Tangier, Morocco. e-mail: lyhyaoui@gmail.com

We have also conducted a general analysis of the CLEF1 (Cross Lingual Evaluation Forum) and the TREC2 (Text Retrieval Conference) results. This general study shows that up to now, only the two first questions levels have been covered by the QA research community. The most successful system in the French monolingual task of the CLEF 2006 used intensive Natural Language Processing (NLP) techniques [2] and obtained an accuracy of 68.95%. Other systems [3][4] obtained an accuracy of 52.63% for the Spanish and the French monolingual task without using any NLP techniques to make their systems as independent of the language as possible. In the TREC 2005 (the TREC 2006 proceedings have not been published yet) the questions were harder to analyze because of their division in groups, each group being related to a target and, therefore, anaphora resolution was also needed. The top accuracy of 53.4% [5] was obtained by means of a sophisticated NER system and a statistical method for the answer selection. In [6] an accuracy of 46.4% was obtained using a dependency relation matching technique in the answer extraction module.

The QA task is considered to be difficult both for design and evaluation. The CLEF organizers offered an evaluation platform for many languages but unfortunately the Arabic language is not included among them. The non-existence of a test-bed for Arabic language makes QA even more challenging. However, there have been two attempts to build Arabic QA systems oriented towards: (i) a structured knowledge-based source of information [7]; and (ii) unstructured data [8]. The test-bed of the second QA system was composed of a set of articles of the Raya newspaper. In the evaluation process, four Arabic native speakers were asked to give the system 113 questions and judge the correctness of

1 http://www.clef-campaign.org
2 http://trec.nist.gov
3 http://www.raya.com

The existence of weapons of mass destruction in Iraq?".

The study of the QA task research guidelines [1] reported that there are generally four kinds of questioners where each type represents questions with a certain level of complexity. The easiest questions are those concerning specific facts; for instance: “Who is the king of Morocco?”, “In which city will ICTIS’07 be held?”, etc.; and the hardest ones need a system able to deduce and decide by itself the answer because the question might be something like “Is there any evidence of the existence of weapons of mass destruction in Iraq?".
its answers manually. The reported results of precision and recall were of 97.3%. These (possibly biased) results seem to be very high if compared with those obtained before for other languages in the CLEF 2006 and TREC 2005 competitions. Unfortunately, the test-bed which was used by the authors is not publicly available in order to compare the QA system with it. In this paper we describe the components of the new ArabiQA system we have been developing. The rest of this paper is structured as follows: In the second section we describe the generic architecture of the system. Special focus is put on named entity recognition in Arabic in section Three. The implementation of the passage retrieval module is presented in section Four. Section Five describes our first attempt for the answer extraction together with the results of the preliminary experiments we carried out. Finally, we draw some conclusions and we discuss the future work to be undertaken, in order to fully implement the ArabiQA system.

II. ARABIQA GENERIC ARCHITECTURE

The generic architecture illustrated in Figure 1 has been adopted to design a QA system oriented to unstructured data. From a general viewpoint, the system is composed of the following components: (i) Question Analysis module: it determines the type of the given question (in order to inform the AE module about the expected type of answer), the question keywords (used by the passage retrieval module as a query) and the named entities appearing in the question (which are very essential to validate the candidate answers); (ii) Passage Retrieval module: it is the core module of the system. It retrieves the passages which are estimated as relevant to contain the answer (see section Four for more details); (iii) Answer Extraction module: it extracts a list of candidate answers from the relevant passages (see section Five for more details); (iv) Answers Validation module: it estimates for each of the candidate answers the probability of correctness and ranks them from the most to the least probable correct ones.

The first, third and fourth modules need a reliable Named Entities Recognition (NER) system. In our case, we have used a NER system that we have designed ourselves [15] (see section Three for more details).

III. NAMED ENTITIES RECOGNISER

A NER system identifies proper names, temporal and numeric expressions in an open-domain text. Such a system is needed in many NLP tasks (IR, QA, Information Extraction, clustering, etc...) because named entities represent for many languages 10% of the available articles [15].

As we mentioned before (see Section 2), a reliable Arabic NER system is needed for most of the components of ArabiQA. Unfortunately, all of the NER systems known in the literature were built for commercial ends (Siraj4 by Sakhr, ClearTags5 by ClearForest, NetOwlExtractor6 by NetOwl and InxightSmartDiscoveryEntityExtractor7 by Inxight). This reason motivated us to build our own Arabic NER system.

The absence of capital letters in the Arabic language makes the NER task even more challenging. To tackle this problem we made a general study of the approaches which proved to be successful for the NER task. In the language-independent NER task of the conference CONLL 20028 the best participations used a Maximum Entropy (ME) approach [16][17][18][19]. A comparison between HMM and ME [20] shows that the ME approach gives better results for the NER task. Moreover, [21] in the NAACL/HLT 20049 reports a language-independent NER system performing on English, Chinese and Arabic texts using a ME approach reached a F-measure of 38.5 for the Arabic language. The corpora used for the training and the test is held by the Language Data Consortium10 (LDC).

Our Arabic NER system is also based on the ME approach. This approach tackles the problem better than others because of its features-based model. For training and testing we have manually built our own corpora which are composed of more than 150,000 tokens with an IOB2 annotation scheme [22]. These corpora are freely available on our website11.

---

4 http://siraj.sakhr.com/
5 http://www.clearforest.com/index.asp
6 http://www.netowl.com/products/extractor.html
8 http://www.cnts.ua.ac.be/conll2002/ner/
9 http://www1.cs.columbia.edu/~pablo/hlt-naacl04/
10 http://www.ldc.upenn.edu
11 http://www.dsic.upv.es/~ybenajiba
For the proper names recognition we have chosen the exponential model where the probability of a word $x$ of being of a class $c$ can be expressed as:

$$p(c|x) = \frac{1}{Z(x)} \cdot \exp \left( \sum_i \lambda_i f_i(x, c) \right)$$  \hspace{1cm} (1)

$Z(x)$ is for normalization and may be expressed as:

$$Z(x) = \sum_{c'} \exp \left( \sum_i \lambda_i f_i(x, c') \right)$$  \hspace{1cm} (2)

Where $c$ is the class, $x$ is a context information and $f_i(x, c)$ is the $i$-th feature. The features are binary functions indicating how the different classes are related to one or many pieces of information about the context in which the word $x$ appeared.

The recognition of temporal and numeric expressions is totally based on patterns and a small dictionary containing the names of days and months in Arabic, as well as numbers written in letters. A more detailed description of our Arabic NER system goes beyond the scope of this paper. For further information, please see [14].

IV. PASSAGE RETRIEVAL

The Passage Retrieval (PR) module is a core component in a QA system. In [9] the authors report that the quality of a QA system depends mainly on the quality of the PR module it uses. In ArabiQA we have tuned the Java Information Retrieval System\(^\text{12}\) (JIRS) in order to be able to process Arabic text [10].

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig2.png}
\caption{JIRS Architecture}
\end{figure}

\[^{12}\text{http://jirs.dsic.upv.es}\]

The JIRS is a multi-platform and language-independent system. It is based on a density approach because it was proved to be the most successful technique for the PR task [11][12][13] (see Figure 2).

The JIRS uses a Distance Density model to compare the n-grams extracted from the question and the passage to determine the relevant passages. The idea of this model is to give more weight to those passages where the question terms appear near to each other. In order to achieve this result JIRS performs in two steps. In the first step, it searches the relevant passages and assigns a weight to each one of them. The weight of a passage depends mainly on the relevant question terms appearing in the passage. The weight of a passage can be expressed as:

$$w_k = 1 - \frac{\log(n_k)}{1 + \log(N)}$$  \hspace{1cm} (3)

Where $n_k$ is the number of passages in which the associated term to the weight $w_k$ appears and $N$ is the number of the system passages. The second step uses a model which gives more importance to passages where the question n-grams appear near to each other. This model can be expressed as:

$$\text{Sim}(p, q) = \frac{1}{\sum_i w_i} \cdot \sum_x h(x) \cdot \frac{1}{d(x, x_{\text{max}})}$$  \hspace{1cm} (4)

Where $x$ is an n-gram of $p$ formed by $q$ terms, $w_i$ are the weights defined by (3), $h(x)$ can be defined as:

$$h(x) = \sum_k w_k$$  \hspace{1cm} (5)

and $d(x, x_{\text{max}})$ is the factor which expresses the distance between the n-gram $x$ and the n-gram with the maximum weight $x_{\text{max}}$, the formula expressing this factor is:

$$d(x, x_{\text{max}}) = 1 + k \ln (1 + D)$$  \hspace{1cm} (6)

Figure 3 shows an illustrating example where the first
passage is more relevant for JIRS than the second one because the keywords *capital* and *Morocco* appear nearer to each other (D=0 in the first passage whereas D=4 in the second one).

In this section, we describe an AE module oriented to Arabic text for only factoid questions. Our system performs in two main steps: (i) the NER system tags all the named entities (NE) within the relevant passage; (ii) the system makes a pre-selection of the candidate answers eliminating NE which do not correspond to the expected type of answer; (iii) the system decides the final list of candidate answers by means of a set of patterns. Figure 4 shows an illustrating example.

The test of the AE module has been done automatically by a test-set that we have prepared specifically for this task. This test-set consists of: (i) list of questions from different types; (ii) list of question types which contains the type of each of the test questions; (iii) list of relevant passages (we have manually built a file containing a passage which contains the correct answer for each question); (iv) and finally a list of correct answers containing the correct answer for each question. We have used manually selected relevant passages in order to estimate the exact error rate of the AE module. The measure we used to estimate the quality of performance of our AE module is the precision (Number of correct Answers/ Number of Questions).

Using the method we described above we have reached a precision of 83.3%.

---

V. Answer Extraction

The AE task is defined as the search for candidate answers within the relevant passages. The task has to take into consideration the type of answers expected by the user [23], and this means that the AE module should perform differently for each type of question. Using a NER system together with patterns seems to be a successful approach to extract answers for factoid questions [8][24][25]. However, for difficult

questions it is needed a semantic parsing to extracts the correct answer [26][27][6]. Other approaches suggest using a statistical method [28].

---

13 http://ar.wikipedia.org
14 http://www.dsic.upv.es/~ybenajiba

---

VI. CONCLUSION AND FUTURE WORK
In this paper, we have described the generic architecture chosen for *ArabiQA* (the Arabic QA system that we have been developing). We particularly focused on the first attempt to implement a simple AE module dedicated especially to factoid questions. In order to implement this module, we developed
an Arabic NER system and a set of patterns for each type of questions (elaborated by hand). To make an automatic test we have manually built a special test-set for this task. We have reached a precision amounting to 83.3%, which shows that the technique we used allows for the efficient extraction of a list of possible answers to a factoid question. In the next future we plan to improve this module in order to extract answers for more difficult questions to analyse than the factoid ones. We are not able to give the accuracy of the complete QA answering system in this paper because some of its components are still in the implementation phase.

The final goal is to complete the implementation of the Question Analysis and Answer Validation components of the ArabicQA system.

REFERENCES


