Arabic Question Answering

A research project report presented

by

Yassine Benajiba

Supervised by

Dr. Paolo Rosso

to

The Department of Information Systems and Computation

in partial fulfillment of the requirements

for the obtention of

Diploma of Advanced Studies

(Diploma de Estudios Avanzados)

in the subject of

Pattern Recognition and Artificial Intelligence

Technical University of Valencia

Valencia, Spain

September 2007

Abstract

The Question Answering (QA) task is a field of Natural Language Processing (NLP) in which researchers try to find ways to make investigation easier for web users looking for a specific information and not just a relevant document. In the literature, great efforts which have been made to build a reliable QA system are reported for most languages. However, attempts to investigate the implementation of a QA system oriented to the Arabic language are scarce. The research work which we present in this document describes with details the methods which proved to be efficient for each the deveoped parts of *ArabiQA*: a QA system fully oriented to the Arabic language and which we hope to release soon.

In order to develop the different parts of the system mentioned above, we have done the following investigations:

1. A study of the Arabic language from a Natural Language Processing viewpoint and a determination of the different peculiarities to be taken into consideration during the development of ArabiQA.

2. The development of an Arabic corpus to test the Arabic-JIRS text passage retrieval system.

3. The adaptation of the JIRS passage retrieval system in order to retrive passages from Arabic text. Taking into consideration the data sparseness of Arabic text, which hardens all the NLP tasks, we have succeeded to enhance significantly the performance of the JIRS system retrieving passages from Arabic text, and thus obtain a robust and efficient Arabic passage retrieval system with a 69% of coverage and 3.28 of redundancy.

4. The development of the annotated *ANERcorp* corpus and lexical resources for training and test of a Named Entity Recognition system.

5. The development of the *ANERsys* Named Entity Recognition system for Arabic text based on the Maximum Entropy approach.

6. The enhancement of the Named Entity Recognition system (*ANERsys 2.0*) by adopting a 2-step approach. Where the first step aims only at detecting the boundaries of the named entities existing in the text. Whereas the task of classifying these entities is left to the second step. The major part of our investigation was dedicated to the development of ANERsys. This task becomes particularily hard for the Arabic language due to some of its particularities.

   We have used for the baseline a script published in the CONLL 2002 and 2003 which gave an F-measure of 43.36. In the first version of our system we have reached an F-measure of 55.23 by using a Maximum Entropy approach and external lexical ressources which have developed ourselves. Furthermore, in the

second version of our system we have adopted a 2-step approach which helped to raise almost 10 points above the previous version.

7. The development of an Arabic corpus which is composed of a list of questions, the passages which contain the good answer and the list of the correct answers in order to test an Answer Extraction module.

8. The development of an Answer Extraction module for Arabic text for Factoid Questions (Who, where and when questions). The Answer Extraction module represents one of the most important parts of a QA system. Moreover, it needs to be built differenlty for each class of questions.

This document presents the different experiments we have carefully conducted to develop each part of the ArabiQA system and describes the future work we have planned in order to complete our system.

Contents

Candidate's Publications

Large portions of this document have appeared in the following papers:

P. Rosso, A. Lyhyaoui, J. Pearrubia, M. Montes-y-Gómez, Y. Benajiba and N. Raissouni. Arabic-English Question Answering. In Proc. of ICTIS-2005. Pages 36–41. 2005.

P. Rosso, Y. Benajiba and Abdelouahid Lyhyaoui. Towards an Arabic Question Answering System. In Proceedings of SRO4, 2006.

Y. Benajiba, P. Rosso and J. M. Gómez Soriano. Adapting the JIRS Passage Retrieval System to the Arabic Language. In A. F. Gelbukh, editor, CICLing 07, volume 4394 of Lecture Notes in Computer Science, pages 530–541. Springer- Verlag, 2007.

Y. Benajiba, P. Rosso and J. M. Benedí. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In A. F. Gelbukh, editor, CICLing 07, volume 4394 of Lecture Notes in Computer Science, pages 143–153. Springer- Verlag, 2007.

Y. Benajiba, P. Rosso and Abdelouahid Lyhyaoui. Implementation of the ArabiQA Question Answering System's Components. In Proceedings of ICTIS 2007.

Y. Benajiba and P. Rosso. Towards a Measure for Arabic Corpora Quality. In Proceedings of CITALA 2007.

Buscaldi D., Benajiba Y., Rosso P., Sanchis E. The UPV at QA@CLEF 2007. In Proc. 8th Int. Cross-Language Evaluation Forum CLEF-2007 working notes, Budapest, Hungary, September 19-21.

Y. Benajiba and P. Rosso. ANERsys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Approach with POS-tag Informationb. In Proceedings of Natural Language-Independent Engineering workshop, 3rd Indian International Conference on Artificial Intelligence, 2007.

Chapter 1

Introduction

Nowadays, the Web has become the main source of information as nearly all kind of data (digital libraries, newspapers collections, etc.) stored in electronic format. The data available is likely to satisfy most requests, nevertheless without the appropriate search facilities, the great amount of retrieved information is practically useless. In fact, the mechanisms developed up to now in Information Retrieval (IR) - those used for instance, by search engines such as Google[1], Yahoo[2] or MSN[3] - allow a user only to retrieve the relevant documents which (partially) match a given query [6]. It is the users task to look for the information within the relevant documents themselves once they are retrieved. In recent years, the combination of the web growth and the explosive demand for better information access has motivated the interest in QA systems [14]. The goal of a QA system is to provide inexperienced users with a flexible access to information allowing them for writing a query in natural language and obtaining not the documents which contain the answer, but the concise answer itself. For instance, given the users question *When was the Technical University of Valencia established?*, we do not want a search engine to retrieve relevant *snippets* containing a link to a document to extract the information from; we would prefer instead a QA system which could instantly return the exact answer: *1971*.

1.1 Question Answering

The QA task is a complex and challenging task both for building a QA system and for evaluating it [14]. TREC[4](Text REtrieval Conference) and CLEF[5](Cross Language Evaluation Forum) are two international competitions allowing to the researchers in this area to compare their systems. Both Monolingual and Cross-Lingual

[1]http://www.google.com
[2]http://www.yahoo.com
[3]http://www.msn.com
[4]http://trec.nist.gov/
[5]www.clef-campaign.org/

QA tasks were organised in these competitions. However, in this paper we will be concerned mainly by the monolingual task. The best accuracy in the monolingual task in CLEF 2006 was 68.95% achieved by [34] for the French language using intensive Natural Language Processing (NLP) techniques both in the indexing step and in the answer extraction module. The second position, with 52.63% was for the Spanish language. [43] used mainly lexical pattern matching and statistical approaches which made their system more independent from the language than the previous one. The third position was for [24] also for the French language. They have also adopted a statistical approach. It is also reported that in order to answer factoid questions their QA system relied mainly on the information provided by the Named Entities Recognition (NER) module, and they also report the necessity of a NER system of high performance for questions such as What and Which ones. On the othe hand, in the TREC competition (which concerns only the Enlglish language) the systems adopted by the participants were more complex than the ones seen in CLEF. The questions are harder to analyze because they are related to a common given target [49]. Therefore, a good anaphora resulotion is needed. The proceedings of the TREC 2006 have not been published yet. For this reason, in this document we only report systems that gave good results in the TREC 2005. [30] obtained the best score in the TREC 2005 with 53.4%. They used a syntactical parser and a NER system as tools accessible to improve the performance of the system, whereas for the answer selection they used a statistical approach. This system has the peculiarity of using a module named logical prover which uses semantic information to proof the correctness of the answer. [46] obtained the second position in the TREC 2005, with an accuracy of 46.4%. The authors report that the good results obtained with this system are due to the dependency relation matching technique used in the answer extraction module. Finally, [16] obtained the third position with 24.6%. This other system adopted a multi-agent structure, with each agent relying on a different QA approach and then, at the end, a combination technique is used to combine all the answers and produce one final answer of the system.

1.2 Arabic Question Answering

In the CLEF and TREC conferences, the participating QA systems were systems performing on many languages (English, French, Spanish, Italian, Dutch, ...) but unfortunately, the Arabic language was not one of them. However, some efforts were conducted to build QA systems oriented to the Arabic language. In [39], a knowledge-based QA system is described; unfortunately, in the paper no results are shown and the system has a quite special architecture since answers are extracted from a knowledge-base (structured data). Moreover, in [29] a QA system based on the 3-module generic architecture (question analysis, passage retrieval and answer extraction), which is adopted by most of the QA systems, is illustrated. For the

test, four native Arabic speakers with university education presented 113 questions to the system and judged *themselves* whether the answer of the system was correct or not. The author reports a precision and a recall reaching 97.3%. However, as we mentioned above, there are no Arabic QA tasks which provide a test-bed allowing a general test for any Arabic QA system, so the reliability of the reported results keep on being very low since such precision and recall were not achieved in any other language.

In this document, we present the first steps of building *ArabiQA*: an Arabic QA system obeying to the general norms reported in the CLEF conference. The second Chapter of this documents gives an overview of what Arabic NLP is and gives a description of its challenges. Whereas the third Chapter describes the different modules of the ArabiQA which has been already developed, and gives details about the experiments we have carried out and the obtained results. Finally, Chapter Four draws our conclusions and future works.w

Chapter 2

The Arabic Language: Definition and Challenges

2.1 Introduction

The Arabic language is a member of the Semitic languages family and it is the most widely spoken one[1] with almost 300 million of first language speakers. The Arabic language has its own script (written from right to left) which is a 28 letters alphabet (25 consonants and 3 long vowels) with allographic variants and diacritics which are used as short vowels except one diacitic which is used as a double consonant marker. The Arabic script does not support capitalization. Numbers are written from left to right which makes a real challenge for the Arabic text editors to handle words written from left to right and others from right to left in the same line (Figure 2.1 shows an example of an Arabic text).

أُسّست الجَامعة المتعدِّدة التقنيات لفالنسيا سنة 1971.

The Technical University of Valencia was established in 1971.

Figure 2.1: Example of Arabic text

[1] http://en.wikipedia.org/wiki/Semitic_languages

2.2 Arabic Encodings

Another challenge for the Arabic text editors is the encoding, the 2 most commonly used encodings are the following:

1. Windows CP-1256: 1-byte characters encoding supporting Arabic, French, English and a small group of Arabic extended characters;

2. Unicode: 2-byte characters encoding and supports all the Arabic extended characters.

Both of of these encodings are human compatible because they allow to normal users to write, save and read Arabic text. However, many problems might be faced when a program is processing an Arabic text encoded with one of the above mentioned encodings. For this reason, Arabic NLP researchers would rather use the Buckwalter encoding. This encoding is a simple mapping from Arabic letter to Roman letters (Figure 2.2 shows the Buckwalter mapping table). Thus, it is more machine compatible because machines are more prepared to work with Roman letters. Nowadays, the Buckwalter encoding is becoming the most commonly used encoding in the Arabic NLP research community and many Arabic corpora such as Arabic Treebank and Arabic Sematic Labeling task corpus used in SEMEVAL 2007 [2] use this encoding.

2.3 Arabic Morphology

The Arabic language has a very complex morphology because of the two following reasons:

1. It is a *derivational* language: All the Arabic verbs have as a root a three or four characters root verb. Similarly, all the adjectives derive from a verb and almost all the nouns are derivations as well. Derivations in the Arabic language is almost always templatic, thus we can say that: *Lemma = Root + Pattern*. Moreover, in case of a regular derivation we can deduce the meaning of a *lemma* if we know the *root* and the *pattern* which have been used to derive it. Figure 2.3 shows an example of two Arabic verbs from the same category and their derivation using the same root. The same pattern has been used for both derivations.

2. It is also an *inflectional* language: *Word = prefix(es) + lemma + suffix(es)*. The *prefixes* can be articles, prepositions or conjunctions, whereas the *suffixes* are generally objects or personal/possessive anaphora. Both prefixes and suffixes are allowed to be combinations, and thus a word can have zero or more affixes (Figure 2.4 shows an example of the composition of an Arabic word).

ء '	ذ *	ل l
آ |	ر r	م m
أ >	ز z	ن n
ؤ &	س s	ه h
إ <	ش $	و w
ئ }	ص S	ى Y
ا A	ض D	ي y
ب b	ط T	ً F
ة p	ظ Z	ٌ N
ت t	ع E	ٍ K
ث v	غ g	َ a
ج j	ـ _	ُ u
ح H	ف f	ِ i
خ x	ق q	ّ ~
د d	ك k	ْ o

Figure 2.2: Buckwalter mapping table

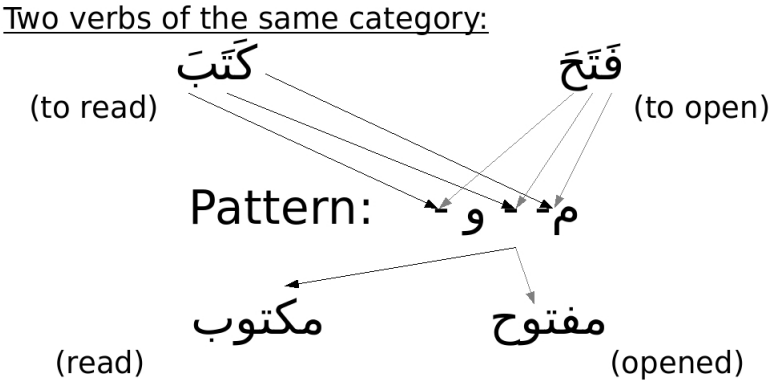


Figure 2.3: An example of Arabic language derivation

However, the Arabic morphology is also very rich. Following we present the morphological features and their possible values for an Arabic verb:

- Aspect : perfective, imperfective, imperative

[2]http://nlp.cs.swarthmore.edu/semeval/tasks/task18/description.shtml

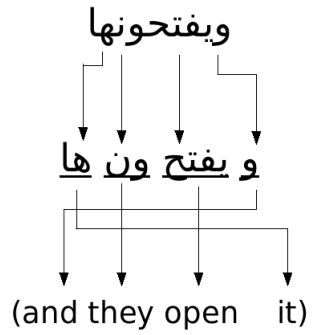


Figure 2.4: An example of Arabic words composition

- Voice : active, passive
- Tense : past, present, future
- Mood : indicative, subjunctive, jussive
- Subject: person, number, gender
- Object : clitic

Moreover, the morphological features for an Arabic noun and their possible values are as follows:

- Number : singular, dual, plural, collective
- Gender : masculine, feminine, neutral
- Definiteness: definite, indefinite
- Case : nominative, accusative, genitive
- Possessive clitic

2.4 Arabic NLP Challenges

2.4.1 Diacritics and ambiguity

As we have mentioned above, diacritics are used in the Arabic language to either put a short vowel or to mark a double consonant. Thus, the same word with different diacritics can express different meanings (see Figure 2.5). Nowadays, diacritics are omitted in all the Arabic texts (books, magazines, newspapers ...) because this allows

Arabic speakers to read faster, and in case of an ambiguous word they can easily disambiguate by using their knowledge and the context in which the word appeared. However, for the Arabic NLP researchers it is one of the main challenges especially for the Word Sense Disambiguation (WSD) and the Machine Translation tasks, and many are the research studies which tempt to diacritize Arabic text [27][50][42][48].

Figure 2.5: An example of the Arabic diaritics influence on the meaning

2.4.2 Absence of capital letters and Named Entity Recognition

In the literature, capital letters are always considered as a crucial characteristic to be used in the recognition of Named Entities (NE's) when this characteristic is supported in the target language. However, it is not the case for the Arabic language (Figure 2.6 shows the example of two words where only one of them is a NE and both of them start with the same character). Thus, the absence of capital letters in the Arabic language is the main obstacle to obtain high peformance in NER [10][7].

Figure 2.6: An example illustrating the absence of capital letters in Arabic

2.4.3 Inflections and data sparseness

From a statistical viewpoint, if Arabic texts are compared to texts written in other languages which have a less complex morphology, the former ones look much more sparse because of the inflectional characteristic of the language that we have mentioned above. It is this specific characteristic of the language which makes more challenging each of the NLP tasks. Following we give an example-based description of some its consequences:

1. *Information Retrieval* and *Question Answering*: The IR task consists of finding the most relevant documents to the query given by the user. Generally, the most relevant documents are those which contain the query keywords. However, if the query keywords appear in a document with additional inflections this document would be classified as irrelevant. For instance, in Figure 2.7 the first portion of the query (maked with a *) appears in the text with the definite article and the second portion (maked with **), which is a proper name, appears with a preposition as a prefix. The only word which appears in the same form in both the query and the text is the the second token of the second portion. Thus, the text will be considered as irrelevant even if it contains the correct information which the user is looking for. Similarly, for the QA task where the user is concerned by an answer to his question, the document containing the answer would not be taking into consideration for the same reason.

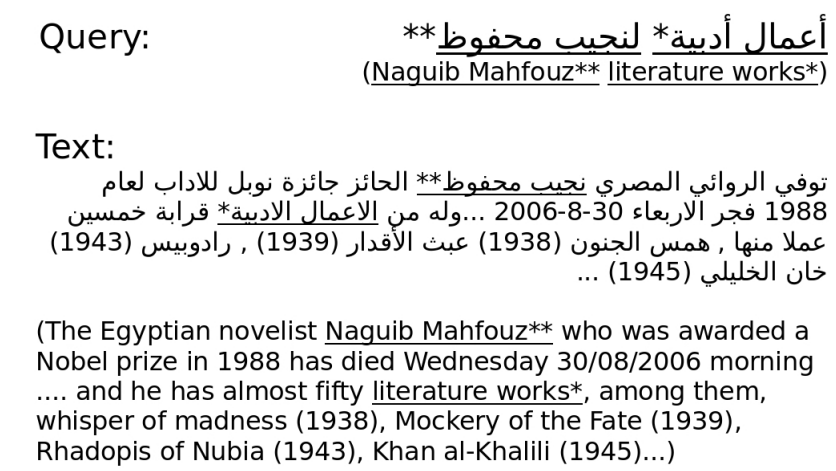


Figure 2.7: An example illustrating how the inflectional characteristic of Arabic hardens the IR task

2. *Information Extraction*: In order to extract a special type of information from an open-domain text, it is very important to take into consideration two prominent features which are: (i) the context in which a word appears; and (ii) the use of additional tools to get more information about a word such as Part-Of-Speech (POS) taggers, Base Phrase (BP) chunkers, etc. However, if the language is highly inflectional (as in the case of Arabic) the same context would appear in different forms, which means that we either need a huge corpus in order to obtain a representative frequency of each of the forms in which a context might appear or find a solution to reduce the number of these forms into a smaller one. Moreover, POS-taggers and BP Chunkers perform very badly

on highly inflectional text. Furthermore, data sparseness makes harder the extraction of information from Arabic text as well (this particular obstacle will be discussed with more details in Section 3.2).

3. *Text Categorization*: In order to decide whether two texts belong to the same category, it is of crucial importance to count the number of words which appear in both documents. Similarly to the tasks which we mentioned above, a comparison between documents is useless if a word has a high probability to appear in a different form in each of its occurrences.

In order to reduce data sparseness in Arabic texts two solutions are possible:

1. *Light stemming*: consists of omitting all the prefixes and suffixes which have been added to a lemma to obtain the needed meaning. This solution is convenient for tasks such as IR and QA because the prepositions, articles and conjunctions are considered as stop words and are not taken into consideration to decide whether a document is relevant for a query or not. An implementation of this solution was available on Kareem Darwish website[3] which has been unfortunately removed.

2. *Word segmentation*: consists of separating the different components of a word by a space character. Therefore, this solution is more adecuate for the NLP tasks which require to keep the different word morphemes such as WSD, NER, etc. A tool to perform Arabic word segmentation trained on Arabic Treebank is available on Mona Diab website[4].

[4] and [8] describe detailed studies of how text segmentation helps to reduce the sparseness in an Arabic document. Furthermore, in the literature different studies show that by reducing data sparseness in Arabic documents their approach allows to obtain a better performance [9][7].

[3] http://www.glue.umd.edu/∼kareem/darwish
[4] http://www1.cs.columbia.edu/∼mdiab/

Chapter 3

The Developed Components of ArabiQA

3.1 ArabiQA Generic Architecture

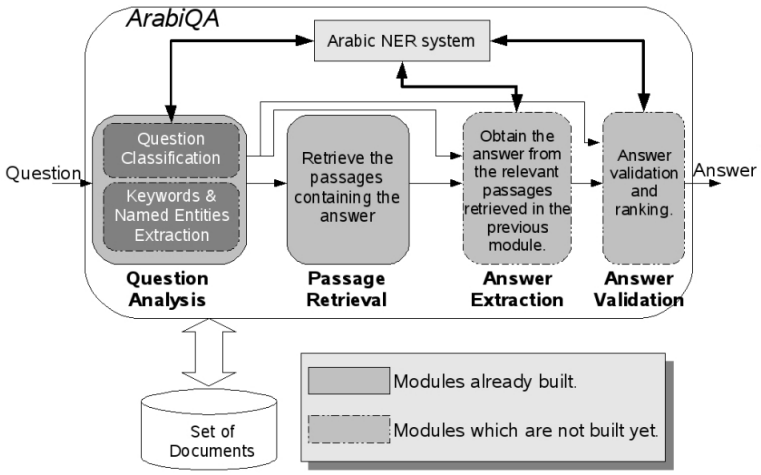


Figure 3.1: ArabiQA generic architecture

The generic architecture illustrated in figure 3.1 is adopted to design a QA system oriented to unstructured data. From a general viewpoint the system is composed of the following components:

1. *Question Analysis* module: it determines the type of the given question (in order to inform the Answer Extraction (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);

2. *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 3.3 for more details);

3. *Answer Extraction* module: it extracts a list of candidate answers from the relevant passages (see section 3.4 for more details);

4. *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 NER system. In our case, we have used a NER system that we have designed ourselves [10] (see section 3.2.4 for more details).

3.2 ANERsys: The Arabic Named Entity Recognition System

3.2.1 Arabic and Language-independent Named Entity Recognition

NE's represent 10% of the articles [23]. Many are the tasks which rely on the huge quantity of information NER systems may provide: Information Extraction (IE), Information Retrieval (IR), Question Answering (QA), text clustering, etc. In the sixth Message Understanding Conference (MUC-6)[1] the NER task was defined as three sub-tasks: ENAMEX (for the proper names), TIMEX (for temporal expressions) and NUMEX (for numeric expression). The first sub-task is the one we are concerned about. ENAMEX was defined as the extraction of proper names and classification of each one of them as one of the following categories:

1. Organization: named corporate, governmental, or other organizational entity;

2. Location: name of politically or geographically defined location;

3. Person: named person or family.

Two are mainly the techniques which were used to build NER systems for the Arabic. They are based, respectively, on the use of a set of keywords and special verbs as triggers and a set of rules to extract the proper names [2], and second using

[1] http://cs.nyu.edu/cs/faculty/grishman/muc6.html

a high precision morphological analysis [36]. With respect to language-independent NER systems, many are the research works which were done: in the shared task of the CONLL 2002 and CONLL 2003[2] for testing the English, Spanish and Dutch corpora, most of the best participants used a maximum entropy approach [11][28][19][18], whereas some others prefered to combine morphological and contextual evidence [18]. Moreover, in [33] very good results were obtained using a character level n-gram model and in [37] a comparison made between the HMM (F-measure of 31.87) and the maximum entropy (55.77) (additional features and a collection of first names as external source allow to increase the F-measure, respectively, up to 84.24 and 85.61). Finally, in the NAACL/HLT 2004[3], a NER system based on maximum entropy for the English, Chinese and Arabic languages [22], obtained F-measure 68.5 for Arabic and 68.6 for Chinese. The Arabic corpus used to carry out the experiments had 166.8k tokens, and it was obtained from ACE Evaluation (September 2003), now it is held now by the Language Data Consortium[4] (LDC) and it is not freely accessible. Furthermore, a text segmentation technique was used for the Arabic text to reduce data sparseness mainly because Arabic is a highly inflected language[5]. Thus, through the above study of the different systems we found out that the technique that mainly proved to be efficient for the NER task is the maximum entropy.

Not many are the available corpora for the NER task. For instance, in the CONLL 2002 conference[6] the available corpora were only for the Chinese, English, French, Japanese, Portuguese and Spanish languages [47]. This is the reason why we had to build our own corpora to carry out this work. It is our intention to make the corpora available in order to share it with other researchers interested in carrying out a comparative work on the NER task in Arabic. It is important to point out that some companies have built Arabic NER systems for comercial ends: Siraj[7] (by Sakhr), ClearTags[8] (by ClearForest), NetOwlExtractor[9] (by NetOwl) and InxightSmartDiscoveryEntityExtractor[10] (by Inxight). Unfortunately, no performance accuracy nor technical details have been provided and a comparative study of the systems is not possible.

Subsection 3.2.2 and 3.2.3 describe the approaches we have adopted to develop ANERsys (an Arabic NER system). An overview of the training and test corpora together with the lexical resources we have developed is given in Subsection 3.2.4. Subsection 3.2.5 describes the experiments we have carried out and draws the obtained results. Finally, subsection 3.2.6 gives a further discussion of the obtained results.

[2] http://www.cnts.ua.ac.be/conll2003/ner/
[3] http://www1.cs.columbia.edu/∼pablo/hlt-naacl04/
[4] http://www.ldc.upenn.edu/
[5] http://corporate.britannica.com/nlt/arabic.html
[6] http://www.cnts.ua.ac.be/conll2002/ner/
[7] http://siraj.sakhr.com/
[8] http://www.clearforest.com/index.asp
[9] http://www.netowl.com/products/extractor.html
[10] http://www.inxight.com/products/smartdiscovery/ee/index.php

3.2.2 ANERsys 1.0: A Maximum Entropy Approach

The Maximum Entropy (ME) technique has been successful not only in the NER task but in many other NLP tasks [12][21][45]. Let introduce the ME approach through a simple example. Let us consider the following sentence taken from the Aljazeera English newspaper[11]:

"Sudan's Darfur region remains the most pressing humanitarian problem in the world, the Food and Agriculture Organisation says."

We need to classifiy the word *"Darfur"* as one of the following four classes: (i) *Pers*: proper name of a *Pers*on; (ii) *Loc*: proper name of a *Loc*ation; or (iii) *Org*: proper name of an *Org*anization; (iv) *O*: not a proper name. If we consider that we do not have any information about the word then the best probability distribution is the one which assigns the same probability to each of the four classes. Therefore, we would choose the following distribution:

$$p(O) = p(Pers) = p(Loc) = p(Org) = 0.25 \tag{3.1}$$

because it is the one that less introduces biases of all the possible distributions. In other words, it is the distribution that maximizes the entropy (In this section we mean by *The best probability distribution* the distribution that minimizes the Kullback-Leibler[12] distance measure to the real probability distribution).

Let suppose instead that we succeeded in obtaining some statistical information from a training corpus and that 90% of the words starting with a capital letter (and not being the first word of the sentence) are proper names. Thus, the new probability distribution would be:

$$p(O) = 0.1 \quad \text{and} \quad p(Pers) = p(Loc) = p(Org) = 0.3 \tag{3.2}$$

This example briefly shows how a maximum entropy classifier performs. Whenever we need to integrate additional information it calculates the best distribution which is the one that maximizes the entropy. The idea behind this approach is that the best distribution is obtained when we do not use any other information but the one we had in the training phase, and if no information is available about some classes, the rest of the probability mass is distributed uniformly between them.

In the example, we managed to make the probability distribution calculations because we considered a reduced number of classes, and we also took into consideration simple statistical information about the proper names (generally called *"context information"*). Unfortunately, this is never true for the real cases where we usually have a greater number of classes and a big range of context information. Therefore, a manual calculation of the probability distribution is not possible. Thus, a robust

[11] http://aljazeera.net

[12] http://ar.wikipedia.org/wiki/Kullback-Leibler_divergence

maximum entropy classifiers model is needed. The exponential model proved to be an elegant approach for the problem which uses various information sources, as the following equation illustrates:

$$p(c|x) = \frac{1}{Z(x)} * exp(\sum_i \lambda_i . f_i(x, c)) \tag{3.3}$$

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

$$Z(x) = \sum_{c'} exp(\sum_i \lambda_i . f_i(x, c')) \tag{3.4}$$

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 context information, for example:

$f_j(x,c)= 1$ if *word(x)="Darfur"* and *c=B-LOC, 0 otherwise.*

To each feature there is an associated weight λ_i since each feature is related to a class and thus it may have a bigger or a lower influence in the classification decision for one class or another. The weights are estimated using the General Iterative Scaling (GIS) algorithm, which ensures convergence on the correct weights after a number of iterations [44].

From a general viewpoint, building a maximum entropy classifier consists of the following steps:

(i) by means of observation and experiments to determine a list of characteristics about the context in which named entities usualy appear (generaly not as simple because some of these information proved not to be so useful and it needs to be replaced; therefore, we might return to this step several times to optimise this list);

(ii) to estimate the different weights λ_i using the GIS algorithm.

(iii) to build a classifier which basically computes for each word the probabilities to be assigned to each of the considered classes: $p(B-PERS|w_i)$, $p(I-PERS|w_i)$, etc. using the ME formula and then assigning the class with the highest probability to this word.

3.2.3 ANERsys 2.0: A Combination of Maximum Entropy with POS-tag Information

In this second version of our Arabic NER system we have adopted a two-steps approach. The first step extracts the boundaries of the NE's. The second one classifies each of the NE's delimited in the previous phase (see Figure 3.2).

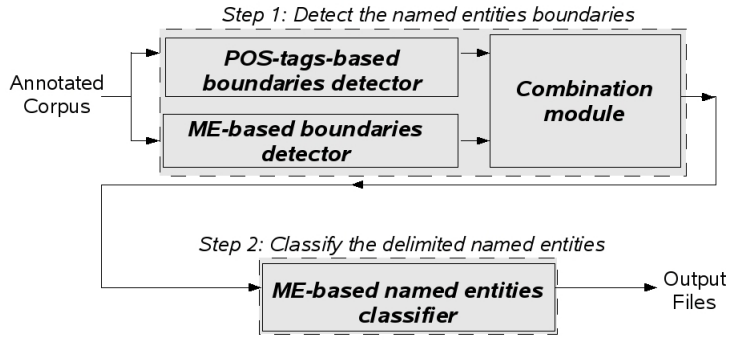


Figure 3.2: Generic architecture of ANERsys 2.0

Step 1 -Named Entities Boundaries Detection As we mentioned above, the first step of our system concerns only the delimitation of the NE's. The input file to this first step should be an IOB2 annotated corpus. The delimitation of the boundaries is made initially by a ME-based and a POS-tag-based modules. Thereafter, the results are combined in a module which was placed at their outputs. Following we present a brief description of the mentioned modules: (i) The ME-based module uses an exponential model which can be illustrated by the following equation:

$$p(c|x) = \frac{1}{Z(x)} * exp(\sum_i \lambda_i . f_i(x, c)) \tag{3.5}$$

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 classes. The λ_i weights are trained using only features related to the beginning and the inside of the NE's and $Z(x)$ is for normalization and may be expressed as:

$$Z(x) = \sum_{c'} exp(\sum_i \lambda_i . f_i(x, c')) \tag{3.6}$$

(ii) On the other hand, for POS-tag-based boundaries detection we have used Mona Diab's Arabic POS tagger. This POS tagger is freely available on her web site[13] in a package together with a tokenizer and a Base Phrase (BP) chunker [**?**]. The author reports that all the tools of the package were trained on data derived from the Arabic Treebank. The model files are included in the package, hence the use of the mentioned tools does not require any type of annotated corpora. These tools were tested on a 400 Arabic sentences and the reported accuracies are very high. To delimit the boundaries we first select the phrases defined as Noun Phrases (tagged as NP) by the BP chunker. The following step is to keep only the NP's whose words

[13]http://www1.cs.columbia.edu/∼mdiab

were tagged as singular, dual or plural proper nouns (tagged as NNP or NNPS by the POS tagger). (iii) Finally, the combination module first conducts a union of the results of the previous two modules. Additionally, a second operation is performed to change the tags which were wrongly put as B-x instead of I-x or vice versa.

Step 2 Delimited NE's Classification The second step of our approach is totally based on ME. We have similarly used the exponential model. The purpose in the second step is to classify each of the NE's delimited in the previous step as one of the four classes which we have mentioned previously (See subsection 3.2.1).

3.2.4 The developed Corpora and Lexical Resources

As we have mentioned in the introduction, it is not possible to find free Arabic corpora oriented to the NER task. Therefore, we have decided to build our own corpora: for training and test. Moreover, we have built also gazetteers to test the effect of using external information sources on the system. It is our intention to make available theses resources on the web in order to ease the further research activity of the NER task in Arabic. Following, we present the main characteristics of the developed resources:

ANERcorp[14]: Two Corpora for Training and Test As reported in the CONLL 2002, the annotated corpora should contain the words of the text together with the correspondent type. The same classes that were defined in the MUC-6 (organization, location and person) were used in the corpora; "Miscellaneous" is the single class that was added for Named Entities which do not belong to any of the other classes. Therefore, any word on the text should be annotated as one of the following tags:

B-PERS	: The Beginning of the name of a PERSon.
I-PERS	: The continuation (Inside) of the name of a PERSon.
B-LOC	: The Beginning of the name of a LOCation.
I-LOC	: The Inside of the name of a LOCation.
B-ORG	: The Beginning of the name of an ORGanization.
I-ORG	: The Inside of the name of an ORGanization.
B-MISC	: The Beginning of the name of an entity which does not belong to any of the previous classes (MISCellaneous).
I-MISC	: The Inside of the name of an entity which does not belong to any of the previous classes.
O	: The word is not a named entity (Other).

In CONLL, it was also decided to use the same format for the training file for all the languages, organising the file in 2 columns: the first column for the words

[14]http://www.dsic.upv.es/∼ybenajiba

and the second one for the tags. Figure 3.3 shows extracts from the training Arabic ANERcorp we developed:

B-PERS	محمود
I-PERS	عباس
O	سيقرر
O	فور
O	عودته
O	من
B-LOC	الاردن
O	نوع
O	الخطوات
O	التي
O	سيتخذها
O	لانهاء
O	الازمة
O	.

Figure 3.3: Extract from the training Arabic ANERcorp

With respect to the CONLL 2002, we have not built three corpora for the Arabic (one for training, another for a first test which consists of fixing parameters and a last one for the final test) but just two corpora (for training and testing). Before, we performed a text normalisation in order to avoid high data sparseness effects. For instance, because of the peculiarity of the language, if no normalisation is performed on the corpus we could find the word *"Iran"* written in two different ways. Unfortunately, the normalisation of the Arabic text is not carried out in a unique way, but looking at the TREC 2001[15] and 2002[8] Arabic/English Cross Lingual IR it is mostly done replacing few characters by an equivalent one. This gave good results for IR systems but it does not seem to be convenient for a NER task because it would cause a loss of valuable information needed to extract the proper names. Therefore, to customise the normalisation definition to our case, in ANERcorp we only reduced the different forms, for instance, of the character *"A"* in just one form.

Finally, we would like to mention that the ANERcorp consists of 316 articles. We preferred not to choose all the articles from the same type and not even from the same newspapers in order to obtain a corpus as generalised as possible. In the following table we present the ratio of articles extracted from each source:

ANERcorp contains 150,286 tokens and 32,114 types which makes a ratio of tokens to types of 4.67. The Proper Names are 11% of the corpus. Their distribution along the different types is as follows:

[15]http://trec.nist.gov/

Table 3.1: Ratio of sources for the extracted article

Source	Ratio
http://www.aljazeera.net	34.8%
Other newspapers and magazines	17.8%
http://www.raya.com	15.5%
http://ar.wikipedia.org	6.6%
http://www.alalam.ma	5.4%
http://www.ahram.eg.org	5.4%
http://www.alittihad.ae	3.5%
http://www.bbc.co.uk/arabic/	3.5%
http://arabic.cnn.com	2.8%
http://www.addustour.com	2.8%
http://kassioun.org	1.9%

Table 3.2: Ratio of phrases by classes

Class	Ratio
PERSon	39%
LOCation	30.4%
ORGanization	20.6%
MISCellaneous class	10%

ANERgazet[16]: Integrating web-based Gazetteers ANERgazet consists of three different gazetteers, all built manually using web resources:

(i) *Location Gazetteer*: this gazetteer consists of 1,950 names of continents, countries, cities, rivers and mountains found in the Arabic version of wikipedia[17];

(ii) *Person Gazetteer*: this was originally a list of 1,920 complete names of people found in wikipedia and other websites. Splitting the names into first names and last names and omitting the repeated names, the list contains finally 2,309 names;

(iii) *Organizations Gazetteer*: the last gazetteer consists of a list of 262 names of companies, football teams and other organizations.

[16]http://www.dsic.upv.es/∼ybenajiba

[17]http://ar.wikipedia.org

3.2.5 Experiments and Results

We have used the ANERcorp (see subsection 3.2.4) to evaluate our system. The baseline model[18] consists of assigning to a word W_i the class C_i which most frequently was assigned to W_i in the training corpus. However, as we have discussed in a previous paper [10], there is no available reference (neither a system nor a corpus) to compare our system with others. For this reason, we have used the demo version of the commercial system Siraj (Sakhr) and converted the obtained files to the IOB2 format to make possible the comparison with our system. We have used the CONLL 2002 evaluation software[19] which considers that a NE is correctly recognised only if: (i) all the constituent words of the NE are recognised; and (ii) the NE is correctly classified. Table 3.3 shows the baseline results. Table 3.4 illustrates the performance of the Siraj (Sakhr) system, whereas Tables 3.5 and 3.6 show the results obtained, respectively, by the first and the second version.

Table 3.3: Baseline results

Baseline	Precision	Recall	F-measure
Location	75.71%	76.97%	76.34
Misc	22.91%	34.67%	27.59
Organisation	52.80%	33.14%	40.72
Person	33.84%	14.76%	20.56
Overall	**51.39**%	**37.51**%	**43.36**

Table 3.4: Siraj (Sakhr) results

Siraj (Sakhr)	Precision	Recall	F-measure
Location	84.79%	67.91%	75.42
Misc	0.00%	0.00%	0.00
Organisation	0.00%	0.00%	0.00
Person	74.66%	55.84%	63.89
Overall	**78.95**%	**46.69**%	**58.58**

[18]http://cnts.ua.ac.be/conll2002/ner/bin/baseline
[19]http://bredt.uib.no/download/conlleval.txt

Table 3.5: ANERsys 1.0 results

ANERsys 1.0	Precision	Recall	F-measure
Location	82.17%	78.42%	80.25
Misc	61.54%	32.65%	42.67
Organisation	45.16%	31.04%	36.79
Person	54.21%	41.01%	46.69
Overall	**63.21**%	**49.04**%	**55.23**

Table 3.6: ANERsys 2.0 results

ANERsys 2.0	Precision	Recall	F-measure
Location	91.69%	82.23%	86.71
Misc	72.34%	55.74%	62.96
Organisation	47.95%	45.02%	46.43
Person	56.27%	48.56%	52.13
Overall	**70.24**%	**62.08**%	**65.91**

3.2.6 Discussion of the Results obtained by ANERsys 2.0

The results show clearly that ANERsys 2.0 performs more than 7 points (F-measure) better than the Siraj (Sakhr) system and significantly better than ANERsys 1.0. However, to make a deeper analysis of the results and have a clearer vision on ANERsys 2.0 we carried out some further experiments. Due to the two-steps approach adopted in the new version of our system we carried out three different tests. A first test to evaluate the performance of the first step of our new approach: i.e., the capacity of the system to delimit the NE's correctly (see Table 3.7). In order to evaluate the exact error rate of the second step, we used a corpus where the NE's delimitations were taken directly from the manually annotated corpus (see Table 3.8).

Table 3.7: Evaluation of the first step of the system

ANERsys 2.0	Precision	Recall	F-measure
B-NE	82.61%	72.10%	77.00
I-NE	91.27%	42.30%	57.81
Overall	**84.27**%	**62.89**%	**72.03**

Table 3.8: Evaluation of the second step of the system

ANERsys 2.0	Precision	Recall	F-measure
Location	93.22%	88.68%	90.90
Misc	94.67%	58.20%	72.08
Organisation	76.89%	65.27%	70.61
Person	75.10%	91.37%	82.44
Overall	**83.22**%	**83.22**%	**83.22**

The results illustrated above clearly that we need to improve the performance of the NE's delimitation process in order to enhance the performance of the complete system. The second step of the system gives an accuracy of 83.22: i.e., in case the first step was perfect the performance of our proposed system would be as good as the the best performance obtained in CONLL 2002 and 2003. Furthermore, it is also important to notice the our system performs better on the Person and Location classes which represent 69.4% of the NE's in our training corpus than the Miscellaneous and Organisation classes which represent only 30.6%. This shows that a greater training corpus will allow us to obtain a better performance.

3.3 Adapting JIRS to the Arabic Language

3.3.1 The JIRS Passage Retrieval System

The PR module is a core component of a QA system. Thus, it was estimated worth to investigate PR modules oriented specifically to QA systems. Those PR modules are more focused on the texts which possibly contain the answer to the user's question than the documents related to the user's query. Many techniques have been investigated in this area. The most successful techniques were the ones based on density [32], [35], [5](JIRS is based on density, see Figure 3.4 and the JIRS architecture description below) and the ones based on terms overlap [13], [20]. However, there are other works which investigated the efficiency of the PR module when the order of the question terms is respected [3] and the possibility of using semantic information to obtain the relevant passages [31].

JIRS is a QA-oriented PR system and it can be freely downloaded from its main web page[20]. As illustrated in Figure 3.4 in order to index the documents the JIRS relies on an n-gram model. To retrieve the relevant passages it performs in two main steps [26]. In the first step it searches the relevant passages and assigns a weight to

[20]http://jirs.dsic.upv.es

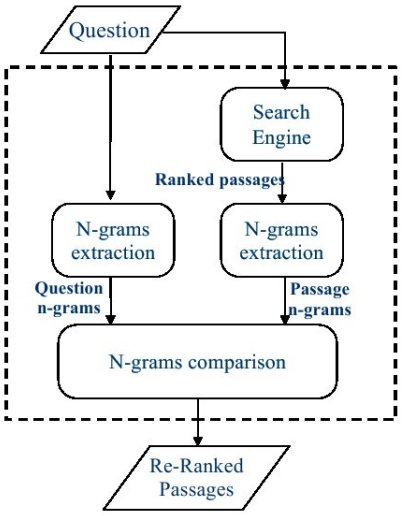


Figure 3.4: The JIRS architecture

each of them. The weight of a passage depends mainly on the relevant question terms appearing in the passage. Thus, the weight of a passage can be expressed as:

$$w_k = 1 - \frac{log(n_k)}{1 + log(N)} \tag{3.7}$$

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 performs only on the top *"m"* passages of the relevant passages returned by the first step (generally m=1000). In this step, JIRS extracts the necessary n-grams from each passage. Finally, using the question and the passage n-grams it compares them using the *Density Distance* model. The idea of this model is to give more weight to the passages where the most relevant question structures appear nearer to each other. For example, let us suppose the question and the two passages shown on Figure 3.5. The correct answer to the question is "Rabat". The Density Distance model would give more weight to the first passage because the distance between the words *capital* and *Morocco* is smaller than the distance between these same words in the second passage.

In order to obtain a bigger weight for the passages that have a smaller distance between question structures, the Distance Density model of a passage p and a question q employes the following equation:

$$Sim(p,q) = \frac{1}{\sum_i w_i} . \sum_x h(x) \frac{1}{d(x, x_{max})} \tag{3.8}$$

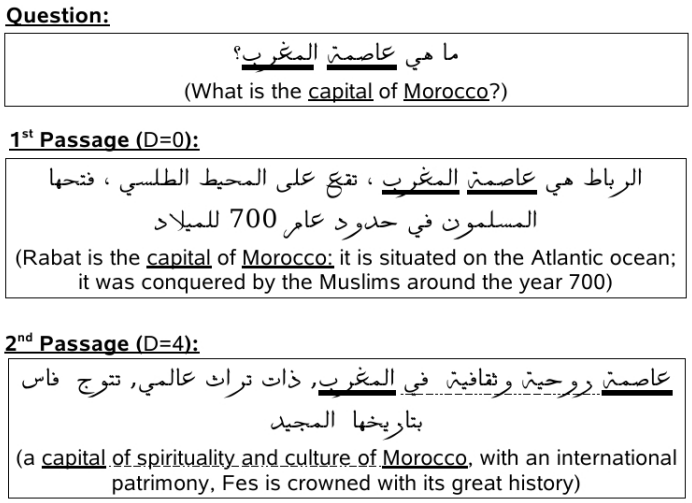


Figure 3.5: An example to illustrate the performance of the Density Distance model (an English translation is given in between parenthesis)

Where x is an n-gram of p formed by q terms, w_i are the weights defined by (1), $h(x)$ can be defined as:

$$h(x) = \sum_k w_k \tag{3.9}$$

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

$$d(x, x_{max}) = 1 + k.ln(1 + D) \tag{3.10}$$

Where D is the number of terms between x and x_{max} (the example given in Figure 3.5 shows an example where D=0 and another where D=4). The last version of the JIRS was reported to perform better than last year in all of the Spanish, French and Italian languages [15]. It was also reported in [15] that the JIRS showed better performance than the Lucene PR system[21] for the Spanish and French languages, whereas the same performance was reported for both systems for the Italian language.

The Arabic-JIRS version of the passage retrieval system relied on the same architecture of Figure 3.4. The main modifications were made on the Arabic language-related files (text encoding, stop-words, list of characters for text normalization, Arabic special characters, question words, etc.). The Arabic-JIRS is also available at the main web page[22].

[21] http://lucene.apache.org/java/docs/
[22] http://jirs.dsic.upv.es

3.3.2 Experiments and Results

Test-bed for Arabic Question Answering[23] In order to test the JIRS on Arabic in the same conditions in which were tested the QA systems which participated in the CLEF 2006 competition we had to develop a test-bed in Arabic with the same characteristics. The test-bed consists of:

(i) *The documents*: we have used a snapshot of the articles of the Arabic Wikipedia[24]. This makes a collection of 11,638 documents. A conversion from the XML to the SGML format was necessary to preprocess the corpus for JIRS;

(ii) *The questions*: we have manually built a set of 200 questions considering the different classes that were reported in the CLEF 2006 competition with the same proportion of each class [25]. These proportions are shown in Table 3.9;

Table 3.9: CLEF 2005 classes Ratio

Class	Number of Questions
NAME	6
NAME.ACRONYM	1
NAME.PERSON	22
NAME.TITLE	1
NAME.LOCATION	6
NAME.LOCATION.COUNTRY	14
NAME.LOCATION.CITY	2
DEFINITION.ORGANIZATION	24
DEFINITION.PERSON	25
DATE	11
DATE.DAY	4
DATE.YEAR	2
QUANTITY	16
QUANTITY.MONEY	3
QUANTITY.DIMENSION	2
QUANTITY.AGE	2
GENERAL	59

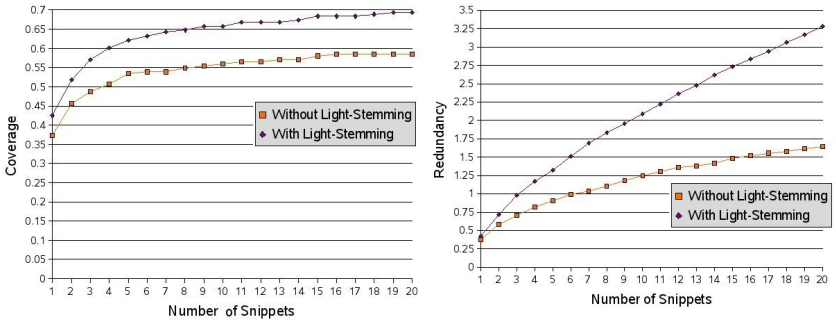
(iii) *The correct-answers*: in order to obtain the *Coverage* (ratio of the number of the correct retrieved passages to the number of the correct passages) and *Redundancy* (average of the number of passages returned for a question) measures automatically from the JIRS, it is necessary to provide, for each of the 200 questions, a list containing

[23] http://www.dsic.upv.es/∼ybenajiba

[24] http://ar.wikipedia.org

all the possible answers. It is also very important to verify that each of these answers is supported by a passage in the collection. We have built the list of the correct-answers and manually verified the existence of each answer in at least one passage of the collection.

Preliminary Results Two experiments have been carried out to estimate the performance of the JIRS on Arabic text. The first expriment consisted of using the test-bed described above. Whereas in the second experiment we performed a light stemming on all the components of the test-bed before we started the retrieval test. The light stemmer we have used for our experiment is the one provided by Kareem Darwish. Figure 3.6 shows the coverage (a) and the redundancy (b) measures for both experiments.



((a)) ((b))

Figure 3.6: Comparison of Coverage and Redundancy of JIRS over both light-stemmed and non-stemmed Arabic corpora

The results presented in Figure 3.6 show that JIRS can retrieve relevant passages also in Arabic, reaching a coverage up to 59% and a redundancy of 1.65 without performing any text preprocessing. However, we carried out a second experiment where we performed a light-stemming to overcome the high data sparesness problem due to the nature of the Arabic language. The light-stemming helped to raise the coverage up to 69% and the redundancy up to 3.28. The values obtained for redundancy show that we cannot reach a higher coverage if we do not use a bigger set of documents.

3.4 Answer Extraction Module: Factoid Questions

The AE task is defined as to search for candidate answers within the relevant passages. The task has to take into consideration the type of answers expected by the user [40], 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 [29][17][1]. However, for difficult questions it is needed a semantic parsing to extract the correct answer [41][30][46]. Other approaches suggest using a statistical method[38]. In this section we describe an AE module oriented to Arabic text for only factoid questions. Our system performs in two main steps:

1. The NER system tags all the named entities (NE) within the relevant passage;
2. The system makes a pre-selection of the candidate answers eliminating NE which do not correspond to the expected type of answer;
3. The system decides the final list of candidate answers by means of a set of patterns.

Figure 3.7 shows an illustrating example.

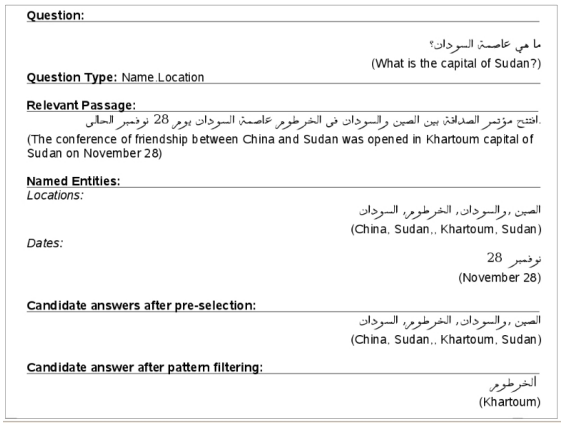


Figure 3.7: Illustrating example of the Answer Extraction module's performance steps

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:

1. List of questions from different types;
2. List of question types which contains the type of each of the test questions;
3. List of relevant passages (we have manually built a file containing a passage which contains the correct answer for each question);
4. List of correct answers containing the correct answer of each question.

We have manually selected relevant passages in order to estimate the exact error rate of the AE module. The measure we have used to estimate the quality of performance of our AE module is precision (Number of correct answers / Number of Questions).

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

Chapter 4

Conclusions and Further Work

The eresearch work carried out led to these following contributions:

4.1 Conclusions

Arabic NLP In this document we have shown most of the challenges which Arabic NLP has. Due to the inflectional characterstic of the Arabic language most of the tasks become harder and most of the techniques require crucial changes and fully new architecture to be Arabic-compatible. We also give examples supported by reliable experiments of the effect of using light-stemming or word segmentation in order to reduce sparseness in Arabic text.

Arabic Named Entity Up to now, we have developed two versions of our NER systems and a training corpus. In the first version we have used Maximum Entropy for classification and an appropriate feature-set which helped together with the gazetteers we have developed to reach an accuracy of 55.23. A deeper anaylsis of our results showed that we have bad results for multiple-word NE's. For this reason, in ANERsys 2.0 we have separated the system into two parts: (i) the first part concerns only detecting the boundaries of the NE's with total ignorance to which class of NE's they might be; (ii) whereas the second step, which receives a corpus where the existant NE's are already identified, aims only at classifying them. This two-step approach allowed us to obtain a performance more than 10 points better than the previous version of the system. However, to have a better view on the performance of the system we have carried out a comparison with the demo version of the Siraj (Sakhr) system in which the commercial Siraj system performed more than 7 points below ANERsys 2.0.

Passage Retrieval system After a complete study of the appropriate technique for the PR task for Arabic text, we found out that using the JIRS PR system, once

adapted to the Arabic text, is a very efficient way to tackle the problem. However, we had to perform a light-stemming or our data in order to boost the results. The results were very promising because we obtained a coverage of 69% and a redundancy of 3.28 using the Arabic Wikipedia as a test corpus which is only composed of 11,000 documents.

4.2 Further Work

Arabic Named Entity Recognition As we have described in Section 3.2, our NER system proved a higher performance in comparison with the Siraj (Sakhr) commercial system. In order to get higher performance we plan in the next future:

1. To use the POS-tag information directly as features for the Maximum Entropy classifier instead of using them separatly in different steps.

2. In order to find out which kind of classification is the most convenient for the NER task we want to try other classifiers than Maximum Entropy such as: Support Vector Machine, Hidden Markov Models, Conditional Random Fields, etc.

3. Using different feature-sets is also a very important experiment. For this reason, we plan to carry out different experiments adding incrementally the different features in order to determine the best feature-set.

4. To use the Arabic Treebank to extract a big lexicon to use as an external resource for our system.

5. As we have noticed, we obtain a better performance for the classes which have bigger number of occurences in the training corpus. We plan to increase the size of the training and test corpora and to give a bigger priority for the classes which occur the least.

Question Classification The first module of ArabiQA has not been developed yet. In order to obtain a reliable Arabic questions classifier we plan to carry out experiments using patterns, descriminative or generative models, as well as combinations of these techniques. A developement of the appropriate training and test corpora is also necessary for a good evaluation of the system.

Answer Extraction As we have mention in Section 3.4, the Answer Extraction module we have developed was focused only on factoid questions. We plan in the next future to enhance this module in order to make it able to answer other types of questions such as definition, list and general questions.

Answer Validation The answer validation module aims at ranking the answers list given by the Answer Extraction module. Moreover, we plan to build this module using web frequencies because we believe that the frequency of occurence of the keywords of the question and the answer in the web is a good indicator to identify weather an answer is good or not to a given question.

Bibliography

[1] S. Abney, M. Collins, and S. Amit. Answer Extraction. In *Proc. of the ANLP 2000*, 2000.

[2] S. Abuleil and M. Evens. Extracting names from arabic text for question-answering systems. *Computers and the Humanities*, 2002.

[3] M. Adriani. Finding Answers to Indonesian Questions from English Documents. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 510–516, Vienna, Austria, 2005. Springer-Verlag.

[4] Y. Almas and A. Khurshid. LoLo: A System based on Terminology for Multilingual Extraction. In *Proc. of Proc. of COLING/ACL-2006 Workshop on Information Extraction Beyond a Document*, pages 56–65, 2006.

[5] C. Amaral, H. Figueira, A. Martins, A. Mendes, P. Mendes, and C. Pinto. Priberams Question Answering System for Poteguese. In Working Notes for the CLEF 2005 Workshop. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 410–419, Vienna, Austria, 2005. Springer-Verlag.

[6] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. New York: ACM Press; Addison-Wesley, 1999.

[7] Y. Benajiba and P. Rosso. ANERsys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-tag Information. In *Proc. of Indian International Conference on Artificial Intelligence*, 2007.

[8] Y. Benajiba and P. Rosso. Towards a Measure for Arabic Corpora Quality. In *Proc. of CITALA-2007*, 2007.

[9] Y. Benajiba, P. Rosso, and J.M. Gómez. Adapting the JIRS Passage Retrieval System to the Arabic Language. In *CICLing 2007 Conference*, volume 4394 of *Lecture Notes in Computer Science*, pages 530–541. Springer-Verlag, 2007.

[10] Y. Benajiba, P. Rosso, and J.M. Benedí Ruiz. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In *CICLing 2007 Conference*, volume 4394 of *Lecture Notes in Computer Science*, pages 143–153. Springer-Verlag, 2007.

[11] O. Bender, F.J. Och, and H. Ney. Maximum Entropy Models For Named Entity Recognition. In *Proc. of CoNLL-2003*, 2003.

[12] J.M. Benedi and F. Amaya. Improvement of a Whole Sentence Maximum Entropy Language Model Using Grammatical Features. In *Association for Computational Linguistics*, 2001.

[13] G. Bouma, J. Mur, G. Van Noord, L. Van Der Plas, and J. Tiedemann. Question Answering for Dutch Using Dependency Relations. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, Vienna, Austria, 2005. Springer-Verlag.

[14] J. Burger, C. Cardie, V. Chaudri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, D. Molodovan, B. Ogdem, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski1, E. Voorhees, and R. Weishedel. Issues, tasks and program structures to roadmap research in question & answering (q&a). Technical report, National Institute of Standards and Technology, Maryland, 2001.

[15] D. Buscaldi, J.M. Gómez, P. Rosso, and E. Sanchis. The UPV at QA@CLEF 2006. In *Working Notes for the CLEF 2006 Workshop*, 2006.

[16] J. Chu-Carroll, K. Czuba, P. Duboue, and J. Prager. IBMs PIQUANT II in TREC2005. In *Proc. of the TREC 2005*, 2005.

[17] R.J. Cooper and S.M. Ruger. A Simple Question Answering System. In *Proc. of the TREC 2000*, 2000.

[18] S. Cucerzan and D. Yarowsky. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proc. of Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pages 90–99, 1999.

[19] J.R. Curran and S. Clark. Language Independent NER using a Maximum Entropy Tagger. In *Proc. of CoNLL-2003*, 2003.

[20] D. Ferrés, S. Kanaan, E. González, A. Ageno, H. Rodríguez, and J. Turmo. The TALP-QA System for Spanish at CLEF 2005. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 400–409, Vienna, Austria, 2005. Springer-Verlag.

[21] M. Fleischman, N. Kwon, and E. Hovy. Maximum Entropy Models for FrameNet Classification. In *Proc. of the 2003 Conference on Emprical Methods in Natural Language Processing*, pages 49–56, 2003.

[22] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. A Statistical Model for Multilingual Entity Detection and Tracking. In *Proc. of NAACL/HLT*, 2004.

[23] N. Friburger and D. Maurel. Textual similarity based on proper names. In *Proc. of MFIR'2002 at the 25 th ACM SIGIR Conference*, pages 155–167, 2002.

[24] L. Gillard, L. Sitbon, E. Blaudez, P. Bellot, and M. El-Béze. The LIA at QA@CLEF-2006. In *Working Notes for the CLEF 2006 Workshop*, 2006.

[25] J.M. Gómez, D. Buscaldi, E. Bisbal-Asensi, P. Rosso, and E. Sanchis. QUASAR, The Question Answering System of the Universidad Politecnica de Valencia. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 439–448, Vienna, Austria, 2005. Springer-Verlag.

[26] J.M. Gómez-Soriano, M. Montes y Gómez, E. Sanchis, and P. Rosso. A Passage Retrieval System for Multilingual Question Answering. In *8th International Conference of Text, Speech and Dialogue 2005 (TSD'05)*, volume 3658 of *Lecture Notes in Artificial Intelligence*, pages 443–450, Karlovy Vary, Czech Republic, 2005. Springer-Verlag.

[27] N. Habash and O. Rambow. Arabic Diacritization through Full Morphological Tagging. In *Proc. of the 8th Meeting of the North American Chapter of the Association of Computational Linguistics Conference*, 2007.

[28] L.C. Hai and T.N. Hwee. Named Entity Recognition with a Maximum Entropy Approach. In *Proc. of CoNLL-2003*, 2003.

[29] B. Hammou, H. Abu-salem, S. Lytinen, and M. Evens. QARAB: A question answering system to support the Arabic language. In *Proc. of the workshop on computational approaches to Semitic languages, ACL*, 2002.

[30] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang. Employing Two Question Answering Systems in TREC-2005. In *Proc. of the TREC 2005*, 2005.

[31] S. Hartrumpf. Extending Knowledge and Deepening Linguistic Processing for the Question Answering System InSicht. In *CLEF 2005, Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 361–369, Vienna, Austria, 2005. Springer-Verlag.

[32] A. Ittycheriah, M. Franz, W.J. Zhu, and A. Ratnaparkhi. IBM's Statistical Question Answering System. In *Proc. of the TREC 2002*, pages 229–234, 2002.

[33] D. Klein, J. Smarr, H. Nguyen, and C. Manning. Named Entity Recognition with Character-Level Models. In *Proc. of CoNLL-2003*, 2003.

[34] D. Laurent, P. Séguéla, and S. Négre. Cross Lingual Question Answer ing using QRISTAL for CLEF 2006. In *Working Notes for the CLEF 2006 Workshop*, 2006.

[35] G.G. Lee, J. Seo, S.Lee, H. Jung, B.H. Cho, C. Lee, B.K. Kwak, J. Cha, D. Kim, J. An, H. Kim, and K. Kim. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In *Proc. of the TREC 2002*, pages 422–451, 2002.

[36] J. Maloney and M. Niv. TAGARAB, A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. In *Proc. of the Workshop on Computational Approaches to Semitic Languages*, 1998.

[37] R. Malouf. Markov Models for Language-Independent Named Entity Recognition. In *Proc. of CoNLL-2003*, 2003.

[38] G.S. Mann. A Statistical Method for Short Answer Extraction. In *Proc. of the ACL-2001 Workshop on Open-domain Question Answering*, 2001.

[39] F.A. Mohammed, K. Nasser, and H.M. Harb. A knowledge based arabic question answering system (aqas). *ACM SIGART Bulletin*, pages 21–33, 1993.

[40] D. Moldovan, M. Pasca, S. Harabagiu, and M. Surdeanu. Performance Issues and Error Analysis in an Open-domain Question Answering System. In *Proc. of the 40th Meeting of the Annual Meeting of the Association of Computational Linguistics*, 2002.

[41] D. Mollá and B. Hutchinson. Dependency-based Semantic Interpretation for Answer Extraction. In *Proc. of the Australian NLP Workshop 2002*, 2002.

[42] R. Nelken and S.M. Shieber. Arabic Diacritization Using Weighted Finite-State Transducers. In *Proc. of the In ACL-05 Workshop on Computational Approaches to Semitic Languages*, pages 79–86, 2005.

[43] M. Pérez-Coutio, M. Montes y Gómez, A. López-López, L. Villaseor-Pineda, and A. Pancardo-Rodríguez. A Shallow Approach for Answer Selection based on Dependency Trees and Term Density. In *Working Notes for the CLEF 2006 Workshop*, 2006.

[44] A. Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania.

[45] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. In *Computer Speech and Language*, pages 187–228, 1996.

[46] R. Sun, J. Jiang, Y. Fan Tan, H. Cui, T. Chua, and M. Kan. Using Syntactic and Semantic Relation Analysis in Question Answering. In *Proc. of the TREC 2005*, 2005.

[47] M.B. Sundheim. Overview of results of the MUC-6 evaluation. In *Proc. of the 6th Conference on Message understanding*, 1995.

[48] D. Vergyri and K. Kirchhoff. Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition. In *Proc. of the COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, 2004.

[49] E. Voorhees. Over TREC 2005. In *Proc. of the TREC 2002*, page 248, 2002.

[50] I. Zitouni, J.S. Sorensen, and R. Sarikaya. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, 2006.