ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy

Yassine Benajiba, Paolo Rosso, and José Miguel Benedí Ruiz

Dpto. Sistemas Informáticos y Computación (DSIC),
Universidad Politécnica de Valencia, Spain
{ybenajiba, prosso, jbenedi}@dsic.upv.es

Abstract. The task of Named Entity Recognition (NER) allows to identify proper names as well as temporal and numeric expressions, in an open-domain text. NER systems proved to be very important for many tasks in Natural Language Processing (NLP) such as Information Retrieval and Question Answering tasks. Unfortunately, the main efforts to build reliable NER systems for the Arabic language have been made in a commercial frame and the approach used as well as the accuracy of the performance are not known. In this paper, we present ANERsys: a NER system built exclusively for Arabic texts based-on n-grams and maximum entropy. Furthermore, we present both the specific Arabic language dependent heuristic and the gazetteers we used to boost our system. We developed our own training and test corpora (ANERcorp) and gazetteers (ANERgazet) to train, evaluate and boost the implemented technique. A major effort was conducted to make sure all the experiments are carried out in the same framework of the CONLL 2002 conference. We carried out several experiments and the preliminary results showed that this approach allows to tackle successfully the problem of NER for the Arabic language.

1 Introduction

We carried out a research on the Arabic language NLP tools and resources in general (corpora, gazetteers, POS taggers,etc). This led us to the conclusion that in comparison with other languages Arabic misses lexical resources, especially free resources available for a research purposes.

Some of the most important resources that any language requires are the NER systems which allow to identify proper names in an open-domain text. The study of English and French newspapers proved that these entities represent 10% of the articles [1]. 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

---

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

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: (i) Organization (named corporate, governmental, or other organizational entity); (ii) Location (name of politically or geographically defined location) or (iii) Person (named person or family). Not many are the available corpora for the NER task. For instance, in the CONLL 2002 conference[2] the available corpora were only for the Chinese, English, French, Japanese, Portuguese and Spanish languages [2]. 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[3] (by Sakhr), ClearTags[4] (by ClearForest), NetOwlExtractor[5] (by NetOwl) and InxightSmartDiscoveryEntityExtractor[6] (by Inxight). Unfortunately, no performance accuracy nor technical details have been provided and a comparative study of the systems is not possible.

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 [3], and second using a high precision morphological analysis [4].

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[7] for testing the English, Spanish and Dutch corpora, most of the best participants used a maximum entropy approach [5][6][7][8], whereas some others prefered to combine morphological and contextual evidence [8]. Moreover, in [9] very good results were obtained using a character level n-gram model and in [10] 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[8], a NER system based on maximum entropy for the English, Chinese and Arabic languages [11], 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[9] (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[10]. Thus, through the above study of the different systems we

---

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

found out that the technique that mainly proved to be efficient for the NER task is the maximum entropy.

The rest of this paper is structured as follows. In the second section of this paper we will focus on the Arabic NER systems. Moreover, the details about Arabic inflections will be given. Section Three will describe with more details the maximum entropy approach. Section Four is dedicated to show the data sets we built to carry out our experimental work. Finally, in the fifth section we present the results of our preliminary experiments, whereas in the sixth section we draw some conclusions and discuss future works.

2   Named Entity Recognition in Arabic

The earlier mentioned language-independent NER systems which participated in the CONLL conference used a general approach based on the common characteristics to all languages. When working with the Arabic language, some important characteristics need to be taken into account:

(i) a character may have up to three different forms, each form corresponds to a position of the character in the word (beginning, middle or end).

(ii) Arabic does not have capital letters; this characteristic represents a considerable obstacle for the NER task because in other languages capital letters represent a very important feature;

(iii) it has long vowels and short vowels, but short vowels are not used anymore in newspapers and this fact introduces a quite high ambiguity in texts (disambiguation using these short vowels is not possible);

(iv) and finally, it is a language with very complex morphology because it is highly inflectional.

The last characteristic is the most important for a NER perspective. The Arabic language is highly inflectional because the general form of a word is:

$$\text{Prefix(es)} + \text{Stem} + \text{Suffix(es)}$$

The number of prefixes and suffixes might be 0 or more. Affixes are added to the stem to obtain the needed expression. For instance, a simple example would be: the word *"manzil"* in Arabic means *"house"* and *"almanzil"* is *"the house"*. This example shows how an Arabic word may be translated in two words. A more complicated example would be, for instance, the word *"wasayaktoubounaha"* which means *"and they will write it"*. If we write this word in the general form introduced above it would be:

$$wa + sa + ya + \text{``ktoub''} + ouna + ha$$

For a NER perpective, this peculiarity of the Arabic language will is a great obstacle because it causes data sparseness.

In the NER system described in [3], a set of rules and keywords was used in order to extract proper names (the problem of data spareseness was not mentioned in the paper). In [11] the authors emphasized this problem and they used an algorithm of text segmentation (introduced in[12]). This algorithm is based on a n-gram language model, and it computes the morpheme trigram probabilities. In order to do so, they have used a manually segmented corpus; it was reported that the algorithm gives an accuracy of 97%. It is important to emphasize that such algorithm is not easy to implement since it requires a large manually segmented corpus for training.

In the ANERsys we take into consideration the data sparsness problem. Instead of performing a text segmentation we use an heuristic method which takes into consideration only prefixes.

3   The Maximum Entropy Approach

The Maximum Entropy (ME) technique has been successful not only in the NER task but in many other NLP tasks [15][16][17]. 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:

$$p(O) = p(Pers) = p(Loc) = p(Org) = 0.25 \qquad (1)$$

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:

$$p(O) = 0.1 \quad \text{and} \quad p(Pers) = p(Loc) = p(Org) = 0.3 \qquad (2)$$

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

---

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

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 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:

$$p(c|x) = \frac{1}{Z(x)} * exp(\sum_i \lambda_i.f_i(x,c)) \tag{3}$$

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

$$Z(x) = \sum_{c'} exp(\sum_i \lambda_i.f_i(x,c')) \tag{4}$$

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 [14].

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.

The feature set we used to implement ANERsys is described in detail in the fifth section.

4   The Developed 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:

4.1   ANERcorp[13]: 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 and the second one for the tags. Figure 1 shows extracts from the CONLL 2002 English training corpus and from the training Arabic ANERcorp we developed:

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[14] and 2002[8] Arabic/English Cross Lingual IR it is mostly done replacing few characters by an

---

[13] http://www.dsic.upv.es/∼ybenajiba
[14] http://trec.nist.gov/

with O	فرانكفورتB-LOC
Del B-PER	, O
Bosque I-PER	اعلنO
in O	اتحادB-ORG
the O	صناعةI-ORG
final O	السياراتI-ORG
years O	فيO
of O	المانياB-LOC
the O	امسO
seventies O	الاولO
in O	انO
Real B-ORG	
Madrid I-ORG	
. O	

Fig. 1. Extracts from the English training corpus used in CONLL 2002 and the training Arabic ANERcorp

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:

Table 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%

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:

Table 2. Ratio of phrases by classes

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

4.2   ANERgazet[15]: 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[16];

(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.

5   Experiments and Results

In order to carry out some experiments we have trained and tested the ANERsys with, respectively, 125,000 and 25,000 tokens of ANERcorp. Furthermore, we used the following feature set which we estimated useful after several experiments (*wi* is the word to classify, *wi-1* is the word appearing before *wi-1* and *wi+1* the word appearing after):

(i) *wi* appears right after a bigram (w_{i-2}, w_{i-1}) or before a bigram (w_{i+1}, w_{i+2}): where (w_{i-2}, w_{i-1}) and (w_{i+1}, w_{i+2}) are elements of a list of bigrams (compiled in the training phase) which usually proper names appear near to;

(ii) *wi* mostly appears in the training phase tagged as class c;

(iii) *wi* is not a stop word (a list of 1650 stop words has been prepared for this feature);

(iv) the class of the previous word is *ci-1*;

(v) *wi*, *wi-1* or *wi+1* are elements of a gazetteer.

We used the *YASMET*[17] software to compute the weights λ_i. First, we used the baseline script[18] to tag each word of the test using a model which consists only of assigning the class which most frequently was assigned to it in the training corpus. And second, we used ANERsys to tag the same test corpus in order

---

to be able to estimate the performance of ANERsys. Furthermore, in order to have a CONLL-like framework, we used the same evaluation software[19]. This evaluation script, accepts as input a file of three columns: the first column contains the words, the second the reference tags and the third the guessed tags. At output it gives the precision, recall and F-measure of each class. Table Three shows the baseline results, whereas Table 4 and 5 illustrate, the performance of ANERsys with and without, respectively, using ANERgazet.

Table 3. Baseline of the ANERcorp test corpus

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 4. ANERsys performance (without using ANERgazet) on the ANERcorp test corpus

ANERsys	Precision	Recall	F-measure
Location	82.41%	76.90%	79.56
Misc	61.54%	32.65%	42.67
Organisation	45.16%	31.04%	36.79
Person	52.76%	38.44%	44.47
Overall	**62.72%**	**47.58%**	**54.11**

Table 5. ANERsys performance (using ANERgazet) on the ANERcorp test corpus

ANERsys	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**

6 Conclusions and Future Works

This paper presents ANERsys, a NER system oriented to the Arabic language, together with ANERcorp and ANERgazet, the resources which were devoloped in the context of the implementation of the system.

In order to carry out the NER task a maximum entropy approach was employed. ME proved to be a convenient solution for the NER task thanks to its

---

[19] http://bredt.uib.no/download/conlleval.txt

feature-based model, and it helped to raise 12 points above the baseline without using any POS-tag information or text segmentation. We investigated also the possibility of integrating web-based gazetteers but we found out that the use of gazetteers does not improve significantly the performance of the system. The same conclusion is supported also by [10], whereas other works [13] showed the contrary. We do not believe that the results did not improve much becuase of the small size of the gazetteers; even so we plan to investigate further this issue.

The main difference observed between the location entities and entities of other classes show that the quality of the system depends mainly on the events seen in the training data because location entities tend to appear in a more regular context than the other entity classes. For this reason, we are planning to increase the ANERcorp training and test corpora in order to obtain better results. In this work we used an ad-hoc method to cope with the data spareseness problem due to the nature of the Arabic language. We plan in the next future to use a more robust algorithm to perform a text segmentation before we train the system. Furthermore, we consider to POS-tag our training and test corpora because it will be a very important feature for a good quality NER system.

Acknowledgments

References

1. Friburger, N., Maurel, D.: Textual Similarity Based on Proper Names. *(MFIR'2002) at the 25 th ACM SIGIR Conference*, Tampere, Finland, 2002, pp. 155–167.
2. Beth M. Sundheim.: Overview of results of the MUC-6 evaluation. In *Proceedings of the 6th Conference on Message understanding*, November 06-08, 1995, Columbia, Maryland.
3. Abuleil, S., Evens, M.: Extracting Names from Arabic text for Question-Answering Systems. *Computers and the Humanities*, 2002 - Springer.
4. Maloney, J., and Niv, M.: TAGARAB, A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, 1998.
5. Bender, O., Och, F. J., Ney, H.: Maximum Entropy Models For Named Entity Recognition. In *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
6. Hai L. Chieu, Hwee T. Ng: Named Entity Recognition with a Maximum Entropy Approach. In *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
7. Curran, JR. and Clark, S.: Language Independent NER using a Maximum Entropy Tagger. In *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
8. Cucerzan, S. and Yarowsky, D.: Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proceedings, 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pp. 90–99.

9. Klein, D., Smarr, J., Nguyen, H., Christopher D. Manning: Named Entity Recognition with Character-Level Models. In *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
10. Malouf, R.: Markov Models for Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
11. Florian, R., Hassan, H., Ittycheriah, A., Jing, H., Kambhatla, N., Luo, X., Nicolov, N. and Roukos, S.: A Statistical Model for Multilingual Entity Detection and Tracking. In *Proceedings of NAACL/HLT*, 2004.
12. Lee, Y-S., Papineni, K., Roukos, S., Emam, O., Hassan, H.: Language Model Based Arabic Word Segmentation. In *Proceedings of the 41st Annual Meeting of the ACL*. pp. 399–406. Sapporo, Japan.
13. Carreras, X., Marquez, L., and Padro, L.: Named Entity Extraction Using AdaBoost. In *Proceedings of CoNLL 2002 Shared Task*, Taipei, Taiwan, September 2002.
14. Ratnaparkhi, A.: A Simple Introduction to Maximum Entropy Models for Natural Language Processing. *Technical Report IRCS-97-08, University of Pennsylvania, Institute for Research in Cognitive Science*.
15. Amaya, F. and Benedi, J.M.: Improvement of a Whole Sentence Maximum Entropy Language Model Using Grammatical Features. *Association for Computational Linguistics*, Toulouse, France, 2001, pp. 10-17.
16. Fleischman, M., Kwon, N., Hovy, E.: Maximum Entropy Models for FrameNet Classification. In *Proceedings of the 2003 Conference on Emprical Methods in Natural Language Processing*, 2003, pp. 49-56.
17. Rosenfeld, R.: A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer Speech and Language*, 10:187228, 1996.