ARNE - A tool for Namend Entity Recognition from Arabic Text

Carolin Shihadeh
DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
carolin.shihadeh@dfki.de

Günter Neumann
DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
neumann@dfki.de

Abstract

In this paper, we study the problem of finding named entities in the Arabic text. For this task we present the development of our pipeline software for Arabic named entity recognition (ARNE), which includes tokenization, morphological analysis, Buckwalter transliteration, part of speech tagging and named entity recognition of person, location and organisation named entities. In our first attempt to recognize named entites, we have used a simple, fast and language independent gazetteer lookup approach. In our second attempt, we have used the morphological analysis provided by our pipeline to remove affixes and observed hence an improvement in our performance. The pipeline presented in this paper, can be used in future as a basis for a named entity recognition system that recognized named entites not only using gazetteers, but also making use of morphological information and part of speech tagging.

1 Introduction

Named entity recognition (NER) is a subtask of natural language processing (NLP). It is the process in which named entities are identified and classified in a text (N. A. Chinchor, 1998). NER is important for NLP, as it supports syntactic analysis of texts and is part of larger tasks, for example information extraction, machine translation or question answering. NLP for the Arabic text is relevant, since Arabic is spoken by more than 500 million people all over the world and there is an enormous number of Arabic sites on the web. The Arabic language has different features that make NLP difficult, such as its complex and rich morphology, the orthographic variation and the non-capitalisation of the Arabic text.

This paper presents a linguistic processing pipeline for Arabic language including tokenization, morphological analysis using a system called ElixirFM developed by Smrz (O. Smrz, 2007), Buckwalter transliteration using the Encode Arabic tool, a placeholder for a part of speech tagger and NER for person, location and organisation named entities. The advantage of such a pipeline model is that the output of one element is the input of the next one, which allows using different resources and information for recognizing named entities. As far as we know, many NER systems combine gazetteers with rules, which consider elements of the surrounding context. In our first approach to recognize named entites from the Arabic text, we have decided to use a gazetteer lookup. A gazetteer is a list of known named entities. If a word is an element in that list then it is labelled as a named entity, otherwise not. The decision of using gazetteers has been influenced by the following criteria:

- Simplicity - developing a NER system which is based on a gazetteer lookup approach is simple.
- Speed - fast execution allow processing large corpora within adequate time.
- Multilingualism - the ability of using the same NER system for any other language, by simply exchanging the used gazetteers.

In our second approach to recognize named entities, we have used the morphological analysis provided by our pipeline to remove affixes such as the conjunction ”wa” and observed therewith an improvement in our performance. The pipeline presented in this paper, can be used in future as a basis for a NER system that recognized named entites not only using gazetteers, but also making use of morphological information and part of speech tagging.

In section 2 of this paper, we describe some related work done on Arabic NER. In section 3, we present our Arabic named entity recognition pipeline software ARNE. In section 4 and 5, ARNE is evaluated and the results are discussed. Finally, in section 6 we give a conclusion and make some suggestions for future work.

2 Related Work

Named entity recognition from Arabic Text has already been studied before. Systems developed in that field can be basically divided into two types: The first type, is based on a handcrafted approach such as the person NER Arabic system PERA and the NER Arabic system NERA, which were developed by Shaalan et al. (2007, 2008). Shaalan et al. used a handcrafted approach in order to create named entity gazetteers and grammars in form of regular expressions, reporting a f-measure of 92,25% resp. 87.5%. Another system that is based on a handcrafted approach, was developed by Elsebai et al. (2009) who used a grammar based approach in which the grammars can be expressed by using an approach called heuristics definition, reporting a f-measure of 89% (Elsebai et al., 2009). Mesfar (2007) used handcrafted syntactic grammars for his Arabic NER system, reporting a f-measure of 87,3% (S. Mesfar, 2007). The second type of systems, is based on a machine learning (ML) approach. Much work on this field was done by Benajiba et al. using different ML approaches such as maximum entropy, conditional random fields and support vector machines, reporting a f-measures of 55.23% - 83.5% (Benajiba et al.). Also, Maloney and Michael Niv used in their system TAGARAB a ML approach, reporting a f-measure of 85.0% (John Maloney and Michael Niv,1998). Nezda et al. used a ML approach to classify 18 different named entity classes, reporting also a f-measure of 85% (Nezda et. al, 2006). During the development of ARNE we have collected information about several named entity recognizers and summarised their most important features in a table. The table can be provided on demand.

3 ARNE System

ARNE (Arabic Named Entity Recognition) is an Arabic NER pipeline system that recognizes person, location and organisation named entities based on a gazetteer lookup approach. In this section of the paper we are going to describe the development of ARNE and explain its architecture. Figure 1 shows the basic architecture. ARNE makes three preprocessing steps before recognizing the named entities: tokenization, Buckwalter transliteration and part of speech tagging. After the preprocessing steps, ARNE performs a named entity recognition, based on a gazetteer lookup approach. In the following four subsections the subtasks of ARNE are introduced.

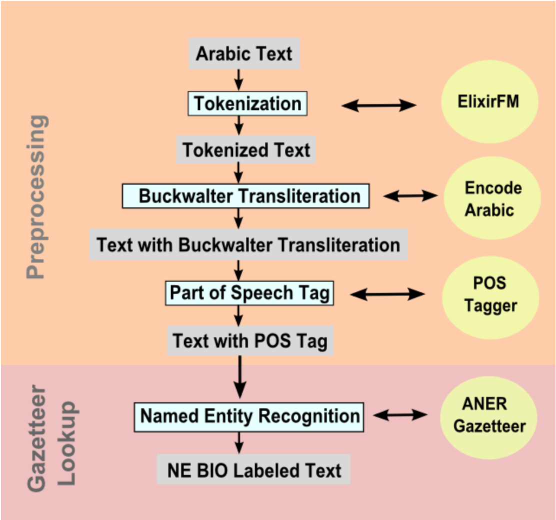


Figure 1: Pipeline architecture of ARNE

3.1 Tokenization

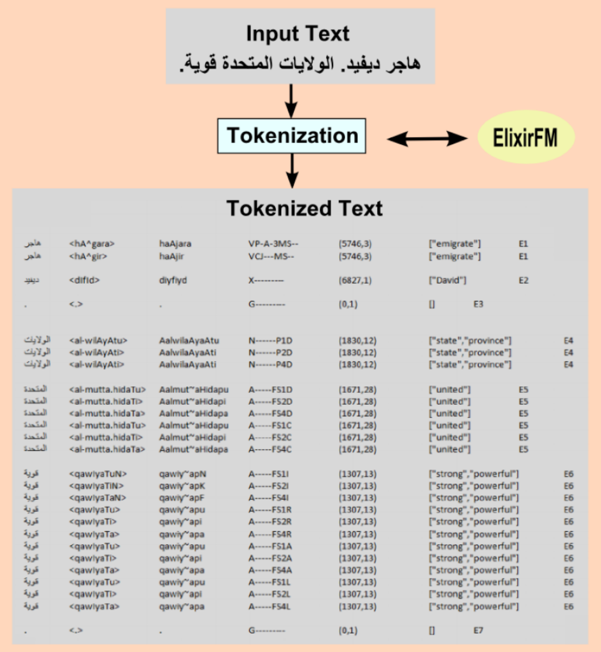
ARNE tokenizes the input text in order to detect the tokens (words, numbers, punctuation marks, special symbols) and sentence boundries. For the Tokenization task in ARNE we have used a system called ElixirFM developed by Smrz (O. Smrz, 2007). The ElixirFM system is able to derive words and inflect them, it can analyse the structure of word forms and recognize their grammatical function. The input of the tokenization task in ARNE is an Arabic text file. This text file is passed to ElixirFM, which outputs a text that contains the following six columns:

- Column 1: Token
- Column 2: ArabTEX notation which indicates both the pronunciation and the orthography
- Column 3: Buckwalter transliteration of the token depending on its pronunciation in column 2. More details to Buckwalter transliteration follow in 3.2
- Column 4: Morphological analysis
- Column 5: Position of the token in the ElixirFM dictionary
- Column 6: English translation

To represent where a token begins and where it ends, each token is written in one line and between one token and the other there is an empty line. To represent where a sentence ends and where the next sentence begins, two empty lines are left between the last token of the first sentence and the first token of the second sentence. After running ElixirFM on the input text and getting the file that contains the previous mentioned six column, ARNE modifies the output file of ElixirFM and adds a seventh column to it: The Elixir Block number, which is a distinct number that identifies each token in the text and serves there with as a pointer to the information obtained by ElixirFM, as ARNE will not save this information again in the forthcoming steps. The features of ElixirFM (column 2-6) and the Elixir Block number is valuable information, but not needed in our gazetteer lookup approach. We can imagine that this information may be of importance for other approaches made for NER or NLP in general.
Figure 2 is an example of the tokenization task in ARNE when inputting the text:

هاجر ديفيد. الولايات المتحدة قوية

Transliterated as: “hAjr dyfyd. AlwlAyAt AlmtHdp qwyp.” and means: “David emigrated. The United States is powerful.”



```
هاجر      <hA^gara>          haAjara           VP-A-3MS--     (5746,3)     ["emigrate"]       E1
هاجر      <hA^gir>           haAjir            VCJ---MS--     (5746,3)     ["emigrate"]       E1

ديفيد     <dIfId>            diyfiyd           X---------     (6827,1)     ["David"]          E2

.         <.>                .                 G---------     (0,1)        []      E3


الولايات  <al-wilAyAtu>      AalwilaAyaAtu     N-------P1D    (1830,12)    ["state","province"]      E4
الولايات  <al-wilAyAti>      AalwilaAyaAti     N-------P2D    (1830,12)    ["state","province"]      E4
الولايات  <al-wilAyAti>      AalwilaAyaAti     N-------P4D    (1830,12)    ["state","province"]      E4

المتحدة   <al-mutta.hidaTu>  Aalmut~aHidapu    A-----FS1D     (1671,28)    ["united"]         E5
المتحدة   <al-mutta.hidaTi>  Aalmut~aHidapi    A-----FS2D     (1671,28)    ["united"]         E5
المتحدة   <al-mutta.hidaTa>  Aalmut~aHidapa    A-----FS4D     (1671,28)    ["united"]         E5
المتحدة   <al-mutta.hidaTu>  Aalmut~aHidapu    A-----FS1C     (1671,28)    ["united"]         E5
المتحدة   <al-mutta.hidaTi>  Aalmut~aHidapi    A-----FS2C     (1671,28)    ["united"]         E5
المتحدة   <al-mutta.hidaTa>  Aalmut~aHidapa    A-----FS4C     (1671,28)    ["united"]         E5

قوية      <qawIyaTuN>        qawiy~apN         A-----FS1I     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTIN>        qawiy~apK         A-----FS2I     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTaN>        qawiy~apF         A-----FS4I     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTu>         qawiy~apu         A-----FS1R     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTi>         qawiy~api         A-----FS2R     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTa>         qawiy~apa         A-----FS4R     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTu>         qawiy~apu         A-----FS1A     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTi>         qawiy~api         A-----FS2A     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTa>         qawiy~apa         A-----FS4A     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTu>         qawiy~apu         A-----FS1L     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTi>         qawiy~api         A-----FS2L     (1307,13)    ["strong","powerful"]     E6
قوية      <qawIyaTa>         qawiy~apa         A-----FS4L     (1307,13)    ["strong","powerful"]     E6

.         <.>                .                 G---------     (0,1)        []      E7
```

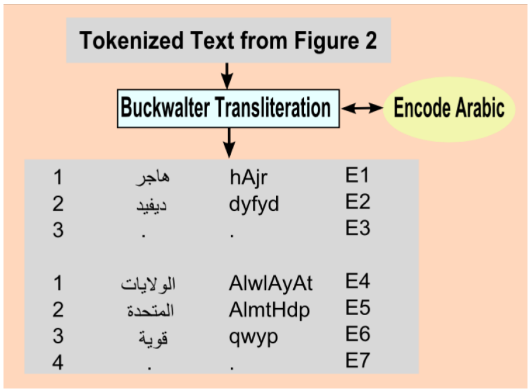
Figure 2: ARNE tokenization of the Text:
هاجر ديفيد. الولايات المتحدة قوية

3.2 Buckwalter Transliteration

We transliterate the Arabic text in order to make it readable for readers who do not have the ability to read the Arabic script but can read the Latin. ARNE uses the Encode Arabic software developed by Tim Buckwalter in order to Buckwalter transliterate the tokens. The input of ARNE in this step is the tokenized text achieved from subsection 3.1. The output is a text that has four columns.

- Column 1: The position of the token in its sentence
- Column 2: The token
- Column 3: The Buckwalter transliteration
- Column 4: The Elixir block number

To represent where one sentence ends and where the other begins we leave an empty line between the sentences and begin numerating the tokens again. As an example for this task, we take the output of Figure 2 as input. The results are illustrated in Figure 3



1	هاجر	hAjr	E1
2	ديفيد	dyfyd	E2
3	.	.	E3
1	الولايات	AlwlAyAt	E4
2	المتحدة	AlmtHdp	E5
3	قوية	qwyp	E6
4	.	.	E7

Figure 3: Buckwalter transliteration of the output of Figure 2

3.3 Part of Speech Tagging

This step is responsible for tagging each word with its part of speech. This task was not implemented yet, but we will integrate our own SVM-based tagger which is based on (Gimenez and Marquez, 2004). Initial evaluation on training and testing with the CoNLL 2006 version of the Arabic dependency treebank yields an 95.38% accuracy. Although this experiment has been performed on properly tokenized and transcribed word forms it is very promising. It is no longer a problem for ARNE, that the POS-tagger is not attached to it yet, as we do not need the POS-tag to recognize named entities in our approach. ARNE performs this step, in order to enable integrating a POS-tagger, which may be useful for other NER approaches. The input of this step is the Buckwalter transliterated text from subsection 3.2. The output adds to the input file a column for the POS-tag, which is at the moment the default value "NULL".

3.4 Named Entity Recognition

In this task the person, location and organisation named entities are labelled using the BIO-labelling method. The overall output of the prepossessing step, comes as a file that contains the tokens, their Buckwalter transliteration, possibly a part of speech tag, which is in ARNE at the moment the default value "NULL", and the "Elixir Block Number". For the NER task, ARNE only needs the Buckwalter transliteration of the tokens, as it goes through the text and looks up the token sequences in the ANERgazet gazetteers developed by Benajiba et al. ARNE uses finite automata in order to handle that task, as it has to define the sequences of tokens that are named entities i.e. the language that contains only words (strings) that are named entities contained in the ANERgazet gazetteer. The reason for using finite automata for the language definition task, is that they are fast simulated by computer, do not use much space and can be generated automatically for example using dk.brics.automaton package. In this subsection we are first, going to describe the finite automata of ARNE. Second, we are going to describe the lookup approach that uses those finite automata in order to label the named entities in the text.

3.4.1 ARNE Finite Automata

In order to recognize named entities, ARNE looks up ANERgazet gazetteers which were developed by Benajiba et al. The ANERgazets consists of three gazetteers (Benajiba, 2005 - 2009):

- Person Gazetteer $Gazet_{Pers}$: This gazetteer contains 2309 names, taken from Wikipedia and other websites.
- Location Gazetteer $Gazet_{Loc}$: This gazetteer consists of 1950 names of countries, cities, mountains, rivers and continents found in the Arabic version of Wikipedia.
- Organisation Gazetteer $Gazet_{Org}$: This gazetteer consists of 262 names of football teams, companies and other organisations.

ARNE contains 3 deterministic, minimised finite automata, each automaton recognizes one of the following languages L:

- Person Language:
  $L_{Pers} := \{w \in \sum^* |w \in Gazet_{Pers}\}$
- Location Language:
  $L_{Loc} := \{w \in \sum^* |w \in Gazet_{Loc}\}$
- Organisation Language:
  $L_{Org} := \{w \in \sum^* |w \in Gazet_{Org}\}$

The alphabet consists of the letters that are used for the Buckwalter transliteration i.e. element of the set {A, b, t, v, j, H, x, d, *, r, z, s, $, S, D, T, Z, E, g,

f, q, k, l, m, n, h, w, y, ’, ¿, &, }, —, {, ’, Y, a, u, i, F, N, K, , o, p, _ , \s}
We have used the dk.brics.automaton java package in order to create for each string in the ANERgazet gazetteers a deterministic finite automaton. After that we merged all those automata to one deterministic finite automaton by creating the power automaton and minimize this automaton using the HOPCROFT algorithm.

3.4.2 ARNE lookup Approach

In this subsection we are going to describe the lookup algorithm used for tagging the tokens with the named entity labels according to the ANERgazet gazetteers, using the BIO-labelling method. A major problem of identifying named entities in text using a gazetter is that named entities are usually multi word entries, especially in Arabic. A simple, but inefficient solution, for extracting the named entities in a text, would be to determine all possible substrings, and match each substring against all gazetters. We will present a more efficient solution, using the morphological analysis provided by our pipeline to remove affixes. The input of ARNE in this step is the POS-tagged text achieved from subsection 3.3. The output is a text that has 6 columns.

- Column 1: The position of the token in its sentence
- Column 2: The token
- Column 3: The Buckwalter transliteration
- Column 4: The POS-tag, at the moment the default value “NULL”
- Column 5: The Elixir block number
- Column 6: The named entity tagb

ARNE looks up strings that have a maximum length of four, because the gazetteers do not contain named entities that consist of more than 4 words. ARNE also assumes that named entities do not cross sentence boundaries, for that reason we handle the named entity labelling task sentence by sentence. The following algorithm, explains how a sentence is labelled using the BIO-labelling method is ARNE.

Lookup Algorithm

```
INPUT:
Sentence $s := t_1 t_2 ... t_n$
Gazetteer $gazet$:  Named entities set

1. Concatenation: For practical
   reasons, concatenate the sentence
   $s$ with the string NULL NULL NULL

   $s' := t_1 t_2 ... t_n NULLNULLNULL$

2. Lookup:

   SET  $i := 1$

   WHILE ( The end of $s'$ is not
   reached ) DO:

     • CASE_1 ( Lookup the string
       $str4 := t_i t_{i+1} t_{i+2} t_{i+3}$ ):

         IF ( $str4 \in gazet$ )
           THEN
             $t_i := B\_NE$
             $t_{i+1} := I\_NE$
             $t_{i+2} := I\_NE$
             $t_{i+3} := I\_NE$
             $i := i + 4$
             GOTO  $CASE_1$
           ELSE
             GOTO  $CASE_2$

     • CASE_2 ( Lookup the string
       $str3 := t_i t_{i+1} t_{i+2}$ ):

         IF ( $str3 \in gazet$ )
           THEN
             $t_i := B\_NE$
             $t_{i+1} := I\_NE$
             $t_{i+2} := I\_NE$
             $i := i + 3$
             GOTO  $CASE_1$
           ELSE
             GOTO  $CASE_3$
```

```
• CASE3 ( Lookup the string $str2 := t_i t_{i+1}$ ):

    IF ( $str2 \in gazet$ )
      THEN
        $t_i := B\_NE$
        $t_{i+1} := I\_NE$
        $i := i + 2$
        GOTO $CASE_1$
      ELSE
        GOTO $CASE_4$

• CASE4 ( Lookup the string $str1 := t_i$ ):
    IF ( $str1 \in gazet$ )
      THEN

        $t_i := B\_NE$
        $i := i + 1$
        GOTO $CASE_1$
      ELSE
        $t_i := O$
        GOTO $CASE_4$

END WHILE
OUTPUT: BIO-labelled sentence $s$
```

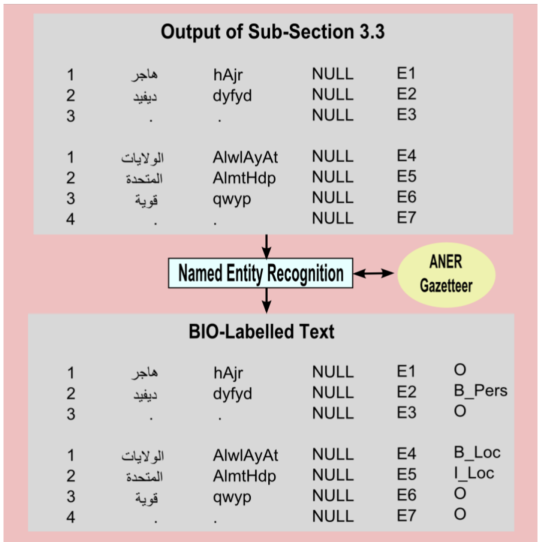
In Figure 4 the task of NE-labelling is illustrated when having the POS-tagged text from Figure 3 as an input.

4 Results

In this section we raise the question of how well ARNE is working in a real application situation. In subsection 4.1 we describe the data used for the evaluation. In subsection 4.2 we present the results of the evaluation.

4.1 Data

For evaluating ARNE, we have used the ANERcorp corpus developed by Benajiba et al. as a goldStandard. The ANERCorp contains more than 150,000 words annotated for the NER task. Since, we use in ARNE the ElixirFM tool for tokenization, we did not have the same tokenization as in the ANERCorp. For the sake of the evaluation, we replaced ElixirFM in ARNE with a tokenizer that simply tokenizes by "white-space" delimiter and got the same tokenization as in the ANERCorp.



1	هاجر	hAjr	NULL	E1
2	ديفيد	dyfyd	NULL	E2
3	.	.	NULL	E3
1	الولايات	AlwlAyAt	NULL	E4
2	المتحدة	AlmtHdp	NULL	E5
3	قوية	qwyp	NULL	E6
4	.	.	NULL	E7

1	هاجر	hAjr	NULL	E1	O
2	ديفيد	dyfyd	NULL	E2	B_Pers
3	.	.	NULL	E3	O
1	الولايات	AlwlAyAt	NULL	E4	B_Loc
2	المتحدة	AlmtHdp	NULL	E5	I_Loc
3	قوية	qwyp	NULL	E6	O
4	.	.	NULL	E7	O

Figure 4: NE-tagged text, when having the POS-tagged text from section 3.3 as an input

4.2 Evaluation

The basic measures for our evaluation are precision, recall and the f-measure. Table 1 summarizes the results of the evaluation. ARNE achieved a f-measure of 30%, which is basically due the small sizes of the gazetteers currently in use. However, we will indicate, how even in this case morphology can help to improve the quality. In section 5, we discuss the results of the evaluation in more detail and make some suggestions for improvements.

ARNE	Precision	Recall	F1 measure
Person	0.1508	0.2100	0.1756
Location	0.6280	0.4498	0.5242
Organisation	0.3744	0.1397	0.2035
Overall	0.3844	0.2665	0.3010

Table 1: Evaluation

5 Discussion

The advantage of using a gazetteer lookup approach for recognizing named entities is that it is simple, fast and language independent. Achieving a f-measure of 30% in our system ARNE, indicates that this approach needs improvement. There are several reasons that the f-measure does not reach higher values, for example the size and the quality of the used gazetteers, the rich and complex Arabic morphology which make the tokenization task to a challenge and finally, the ambiguity problem, which is not considered when using a gazetteer lookup approach. The following subsections explain those problems in more detail and show how a higher f-measure can be achieved by solving some of those problems.

5.1 Gazetteer Size and Quality

The quality of the gazetteers is essential, when using a gazetteer lookup approach. A gazetteer should not contain wrong entries. We went manually through the ANERgazet gazetteers and searched for mistakes. In table 2 we list the wrong entries we have found and mention how often they have occured in the ANERCorp corpus which we have used for evaluating our system.

Word	Meaning	Gazetteer	Occurrence
mn	from	PERS	3188
Alywm	today	PERS	149
AlmADy	the past	PERS	128
AlAwl	the first	PERS	40
wA$nTn	Washington	PERS	48
w	and	LOC	217

Table 2: Occurrence of wrong gazetteer entries

We removed the wrong entries from the gazetteers and evaluated again. This small experiments, improved our f-measure from 30% to 32.5%. Table 3 summarizes the results.

ARNE	Precision	Recall	F1 measure
Person	0.2487	0.2095	0.2274
Location	0.6720	0.4587	0.5452
Organisation	0.3744	0.1397	0.2035
Overall	0.4317	0.2693	0.3253

Table 3: Evaluation using modified gazetteers

Not only the quality of the gazetteers play a fundamental role in achieving good results, but also the size of the gazetteers. Many named entities could not be recognized by ARNE, because they are not part of the ANERGazet gazetteers. The ANERGazet gazetteers have been built by Benajiba, who mentions in his thesis that those gazetteers are very small (Benajiba, 2005 -2009). Another problem is, that different writers and typists have a different point of view how things are orthographically correct or permissible and not all computer platforms and keyboards allow the same symbols (Soudi et al., 2007). If a named entity is written in the corpus differently than in the used gazetteers, then ARNE will not be able to recognize that named entity, since the ANERGazet gazetteers do not cover all the possible writing variants of a word. We assume that expanding the used gazetteers would increase the f-measure. But, we should not forget that any person or organisation gazetteer will probably have poor coverage, since new organisations and new person names come into existence every day.

5.2 Ambiguity

Assuming, we succeed to create a gazetteer that has no mistakes and covers all possible named entities then, we will still have the ambiguity problem, since many named entity terms are ambiguous. A NER system without ambiguity resolution, cannot perform robust and accurate NER.

5.3 The Arabic rich and complex morphology

The Arabic language has a complex and rich morphology because it is highly inflectional. One observation we have made was that ARNE could not recognize phrases like

وسوريا

transliterated as “wswryA” which means “and Syria” and is written as “andSyria”. The named entity “Syria” could not be recognized because the gazetteers contain only the named entity “Syria” and not the phrase “andSyria”. We used the morphological information given by ElixirFM to find out whether a phrase contains a conjunction or not and considered this information in our tagging algorithm. Using this morphological information, our f-measure improved from 32.5% to 33.7%. Table 4 summarizes the results.

6 Conclusion and Future Work

We have presented the development of a pipeline software for Arabic named entity recognition (ARNE), which includes tokenization, morphological analysis, Buckwalter transliteration, a place-

ARNE	Precision	Recall	F1 measure
Person	0.2542	0.2159	0.2335
Location	0.6861	0.1400	0.5769
Organisation	0.3676	0.1400	0.2028
Overall	0.4359	0.1653	0.3377

Table 4: Evaluation using morphological information

holder for a part of speech tagger and named entity recognition of person, location and organisation named entities. We have used a gazetteer lookup approach for recognizing named entities from the Arabic text and achieved a f-measure of 30%. Although this low result are basically due the small number of gazetteers, our system provides easy ways of extending it, which is one of our next focus. We have illustrated the boundaries of a gazetteer lookup approach, such as the incapability of creating gazetteers with full coverage and the inability to treat ambiguity. We have demonstrated with some experiments how this performance can be improved, by using for example the morphological information provided by our pipeline.

As future work we intend to integrate a POS-tagger to ARNE, extend the gazetteers, use the POS-tag information and the morphological information provided by ElixirFM to improve the performance and finally, make our lookup algorithm more efficient using parallel programming.

7 Acknowledgments

We wish to thank Dr. Otakar Smrz not only for his system ElxirFM which we have used in our NER system ARNE, but also for the innumerable emails he has written us and the phone calls we had, making us understand the system ElixirFM more deeply and giving us hints how to attach ElixirFM to ARNE. Our thanks goes also to Dr. Yassine Benajiba, who made his gazetteer and corpus available for us and for supporting us to understand his systems ANERsys. We wish also to thank Dr. Khaled Shaalan, Dr. Nizar Habash, Dr. Slim Mesfar, Dr. Hayssam Traboulsi, Dr. Farid Meziane and all people who answered our questions to their papers and made it possible to create the table that summarises work done on Arabic NER. Finally, we would like to thank Alexander Volokh for beta reading.

References

N. A. Chinchor 1998. *Muc-7 named entity task definition (version 3.5), MUC-7.*

K. Shaalan and H. Raza 2007. *Person name entity recognition for arabic, in Semitic 07: Proceedings of the 2007 Workshop on Computational Ap-proaches to Semitic Languages, Morristown, NJ, USA, 2007, Association for Computational Linguistics, pp. 17-24*

K. Shaalan and H. Raza 2009. *NERA: Named Entity Recognition for Arabic. Journal of the American Society for Information Science and Technology archive Volume 60 Issue 8, August 2009 Pages 1652-166 John Wiley and Sons, Inc. New York, NY, USA*

Elsebai, Meziane, Belkredim 2009. *A Rule Based Persons Names Arabic Extraction System. Communications of the IBIMA Volume 11, 2009 ISSN: 1943-7765*

S. Mesfar 2007. *Named entity recognition for arabic using syntactic grammars, in Lecture Notes in Computer Science, Berlin / Heidelberg, , pp. 305-316*

P. R. Yassine Benajiba and J. M. B. Ruiz *Anersys: An arabic named entity recognition system based on maximum entropy*

Yassine Benajiba. *Arabic Named Entity Recognition, PhD thesis* Universidad Politecnica de Valencia.

Yassine Benajiba and P. Rosso *Improving ner in arabic using a morphological tagger.*

P. R. Yassine Benajiba, Mona Diab *Arabic named entity recognition: An svm-based approach.*

John Maloney and Michael Niv *TAGARAB: A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. SRA International Corp. 4300 Fair Lakes Court Fairfax, VA 22033*

2006 Luke Nezda, Andrew Hickl, John Lehmann, and Sarmad Fayyaz *What in the World is a Shahab? Wide Coverage Named Entity Recognition for Arabic. Language Computer Corporation 1701 N. Collins Blvd. Richardson, TX 75080, USA*

O. Smrz 2007. *Functional Arabic Morphology Formal System and Implementa-tion, PhD thesis, CHARLES UNIVERSITY IN PRAGUE*

A. Soudi, A. van den Bosch, and G. Neuman 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods, Springer Publishing Company, Incorporated.*

Gimenez, J. and Marquez 2004. *Svmtool: A general pos tagger generator based on support vector machines. In In Proceedings of LREC04, vol. I, pages 43 - 46. Lisbon, Portugal, 2004. (ISBN 2-9517408-1-6).*