Person Name Entity Recognition for Arabic

Khaled Shaalan
Institute of Informatics
The British University in Dubai
P O Box 502216, Dubai, UAE
Khaled.shaalan@buid.ac.ae

Hafsa Raza
Institute of Informatics
The British University in Dubai
P O Box 502216, Dubai, UAE
hafsa.raza@gmail.com

Abstract

Named entity recognition (NER) is nowadays an important task, which is responsible for the identification of proper names in text and their classification as different types of named entity such as people, locations, and organizations. In this paper, we present our attempt at the recognition and extraction of the most important proper name entity, that is, the person name, for the Arabic language. We developed the system, Person Name Entity Recognition for Arabic (PERA), using a rule-based approach. The system consists of a lexicon, in the form of gazetteer name lists, and a grammar, in the form of regular expressions, which are responsible for recognizing person name entities. The PERA system is evaluated using a corpus that is tagged in a semi-automated way. The system performance results achieved were satisfactory and confirm to the targets set forth for the precision, recall, and f-measure.

1 Introduction

The recognition and classification of proper names in text (e.g. persons, locations, and organizations) has recently become considered of major importance in Natural Language Processing (NLP) as it plays a significant role in various types of NLP applications, especially in Information Extraction, Information Retrieval, Machine Translation, Syntactic Parsing/Chunking, Question-Answering, among others. The valuable information in text is usually located around proper names, to collect this information it should be found first (Abuleil, 2004; Chinchor, 1998). In our presentation, we will concentrate on the role of NER in Information Extraction (IE). IE is the NLP task that retrieves relevant information from unstructured texts and produces as a result a structured set of data.

This paper describes work on recognizing and extracting the most important entities, that is, person names for the Arabic language. We have adopted the rule-based approach using linguistic grammar-based techniques to develop PERA. This approach provides flexibility and adaptability features in our system and it can be easily configured to work with different languages, NLP applications, and domains. In order to determine the best rules for recognition of person names, various Arabic text corpora were analyzed. Phrases containing person names were retrieved, the underlying pattern was learned and person indicators such as titles were identified. Apart from this, person names were extracted from the available corpora and other resources to build up a lexicon, in the form of gazetteer name lists, or gazetteer for short. The various Arabic naming conventions and the person indicators identified helped in deriving fine rules that gave high-quality recognition of person names in Arabic text. The recognition was done in two cycles using first the gazetteer and then the grammar rules. The PERA system is evaluated using a reference corpus that is tagged with person names in a semi-automated way. The achieved system performance results were satisfactory and confirm

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 17–24,
Prague, Czech Republic, June 2007.

to the targets set forth for the precision, recall, and f-measure.

The paper is structured as follows. Section 2 presents the related work. Section 3 describes the naming conventions of person names used in Arabic language. Section 4 presents methods of data collection used. Section 5 explains the system architecture and implementation. Section 6 presents the experiment performed to evaluate the system and finally Section 7 concludes the paper, summarizes our achievements, and highlights our plans for future work..

2 Related Work

As in other NLP techniques, there are two main approaches to NER (Toral, 2005). One is based on linguistic knowledge, in particular grammar rules and hence called rule-based, while the other is based on machine learning techniques. The required resources for the knowledge approach are usually gazetteers and rules whereas the learning approach needs an annotated (tagged) corpus. The linguistic knowledge-based model achieve better results in specific domains, as the gazetteers can be adapted very precisely, and it is able to detect complex entities, as the rules can be tailored to meet nearly any requirement. However, if we deal with an unrestricted domain, it is better to choose the machine learning approach, as it would be inefficient to acquire and/or derive rules and gazetteers in this case.

Name identification has been worked on quite intensively for the past few years, and has been incorporated into several products. Many researchers have attacked this problem in a variety of languages but only a few limited researches have focused on NER for Arabic text. This is due to the lack of resources for Arabic NE and the limited amount of progress made in Arabic NLP in general.

Maloney and Niv (1998) developed TAGARAB an Arabic name recognizer that uses a pattern-recognition engine integrated with morphological analysis. The role of the morphological analyzer is to decide where a name ends and the non-name context begins. The decision depends on the part-of-speech of the Arabic word and/or its inflections.

Abuleil (2004) presented a technique to extract proper names from text to build a database of names along with their classification that can be used in question-answering systems. This work was done in three main stages: 1) marking the phrases that might include names, 2) building up graphs to represent the words in these phrases and the relationships between them, and 3) applying rules to generate the names, classify each of them, and saves them in a database.

Larkey et al. (2003) have conducted a study that showed the importance of the proper names component in cross language tasks involving searching, tracking, retrieving, or extracting information. In particular, they have concluded that a combination of static proper name (English-Arabic) translation plus transliteration provides a successful solution.

Pouliquen et al. (2005) developed a tool for multilingual person name recognition that focuses on the "Who" part of the analysis of large news text. As multilingual NER is concerned, the transliteration of the NE has included alternative spelling variants where the origin language of the name is usually not known. Several variants could also be found in the same language.

Samy et al. (2005) has used parallel corpora in Spanish, and Arabic and an NE tagger in Spanish to tag the names in the Arabic corpus. For each sentence pair aligned together, they use a simple mapping scheme to transliterate all the words in the Arabic sentence and return those matching with NEs in the Spanish sentence as the NEs in Arabic. While they report high precision and recall, it should be noted that their approach is applicable only when a parallel corpus is available.

Zitouni et al. (2005) has adopted a statistical approach for the entity detection and recognition (EDR). In this work, a *mention* can be either named (e.g. John Mayor), nominal (the president) or pronominal (she, it). An entity is the aggregate of all the mentions (of any level) which refer to one conceptual entity. This extended definition of the entity has proved the suitability of the approach.

3 Components of an Arabic Full Name

Arabic has well-defined naming practices. The Arabic name elements may be divided into five main categories, Ibn Auda (2003):

1. An ***ism*** (pronounced IZM, as the final syllable in the word *dogmatism*), a personal, proper name given shortly after birth, i.e. the given name. Examples of such names are *Muham-*

mad [Mohammed], *Musa* [Moses], *Ibrahim* [Abraham].

2. A ***kunya*** (pronounced COON-yah), an honorific name or surname, as the father or mother of someone; e.g., *abu Da'ud* [the father of David], *umm Salim* [the mother of Salim]. It is meant as a prefix of respect or reverence. Married persons (especially married ladies) are, as a general rule, simply called by their *kunya* (*abu* or *umm* + the name of their first-born child). When using a person's full name, the *kunya* precedes the personal (given) name: *Abu Yusuf Hasan* [the father of Joseph, Hasan], *Umm Ja'far Aminah* [the mother of Ja'far, Aminah].
3. By a ***nasab*** (pronounced NAH-sahb), a pedigree, as the son or daughter of someone; *e.g.*, *ibn 'Umar* [the son of Omar], *bint 'Abbas* [the daughter of Abbas]. The *nasab* follows the *ism* in usage: *Hasan ibn Faraj* [Hasan the son of Faraj], *Sumayya bint Khubbat* [Sumayya the daughter of Khubbat]. Many historical personages are more familiar to us by their *nasab* than by their *ism*: e.g., the historian *ibn Khaldun*, the traveler *ibn Battuta*, and the philosopher *ibn Sina* [Avicenna].
   *Nasab*s may be extended for several generations, as may be noted in the example below containing two generations nasab:
   Abu al-Qasim Mansur **ibn** al-Zabriqan **ibn** Salamah al-Namari
4. A ***laqab*** (pronounced LAH-kahb), a combination of words into a byname or epithet, usually religious, relating to nature, a descriptive, or of some admirable quality the person had (or would like to have); e.g., *al-Rashid* [the Rightly-guided], *al-Fadl* [the Prominent]. *Laqab*s follow the *ism*: *Harun al-Rashid* [Aaron the Rightly-guided].
5. A ***nisba*** (pronounced NISS-bah), a name derived from a person's: trade or profession, place of residence or birth, religious affiliation, among others; e.g. *al-Hallaj* [the dresser of cotton], *Al Msri* [The Egyptian], *Islami* [Islamic]. Nisbas follow the ism or, if the name contains a nasab (of however many generations), generally follow the nasab.

4 Data Collection

The development of the system PERA depends on collecting dictionaries of proper nouns and their related indicators. Techniques used for acquiring such data to build the dictionaries include:

1. *Automatic collection of person names from annotated corpus.* The person entities in the ACE[1] and Treebank corpus[2] were recognized and extracted using regular expression patterns coded within Python scripts. Python is a strong string processing language and widely used in developing NLP applications and tools.
2. *Identification of person indicators.* Apart from extracting the person names, these corpora were used also to extract noun phrases containing the person names. The surrounding sequence of words around person names was analyzed to identify indicators of person names. A dictionary of these indicators was formed which represented contextual cues of person names.
3. *Name Database provided by government organization.* The person name dictionary was also build from names collected from some organizations including Immigration Departments, Educational bodies, and Brokerage companies.
4. *Internet Resources.* Names were retrieved further from various websites[3] containing lists of Arabic names. Some of these names are Romanized (written using the Latin alphabet) and had to be transliterated from English to Arabic. This was done using the online translation software 'Tarjim' provided by Sakhr Software Company. Notice that the variations in Romanized Arabic due to the lack of one to one correspondence between Arabic letters and Roman letters have also been reflected in the transliteration, in reverse, from Romanized Arabic to Arabic Script.

The raw data received had to be further processed to make it suitable for building gazetteers to

[1] ACE reference: http://projects.ldc.upenn.edu/ace/
[2] Treebank Corpus reference:
http://www.ircs.upenn.edu/arabic/
Both software are available to BUiD under license agreement.
[3] Web sites include:
http://en.wikipedia.org/wiki/List_of_Arabic_names ,
http://www.islam4you.info/contents/names/fa.php, and
http://www.mybabynamessite.com/list.php?letter=a

be incorporated within the system. Some of the automated preprocessing performed on these data includes:

- Removing extra whitespaces between first and last names, or beginning and end of names for the efficient processing of the main gazetteer (dictionary) of complete person names.
- Creating separate dictionaries (i.e. first, last and middle names) without redundancy because the full names had to be parsed. The extraction of each of these individual components from full person names was based on Python code and common sense.

4.1 Typographic Variants

In order to be able to recognize variant Arabic name entities, we added extra expressions in rules and lexicon entries which lead to recognizing named entities and their typographic variants. Examples of typographic variants include:

- The drop of hamza initially, medially, and finally (e.g. احسان vs إحسان - [*Ehessan*])
- Two dots inserted on aleph maqsura, and two dots removed from yaa (e.g. موسي vs موسى-[*Mousa*])
- Dropping the madda from the aleph (e.g. ال خليفة vs آل خليفة - [*Al Khalifa*])
- Hamza insertion below vs. above aleph (e.g. أسراء vs إسراء-[*Essraa*])
- Two dots inserted on final haa, and two dots removed from taa marbouta (e.g. فاطمه vs فاطمة-[*Fatma*])
- Diacritics: partial, full, or none. In the current version we remove diacritics.
- Typing hamza followed by aleph maqsura separately vs. together (e.g. هانىء vs هانئ-[*Hani*]).

4.2 Dictionaries

The following dictionaries (gazetteers) are derived using the aforementioned data collection techniques. A total of 472617 entries were collected.

- A dictionary of full person names (263598 entries)
- A dictionary of first names (78956 entries)
- A dictionary of middle names (67595 entries)
- A dictionary of last names (33517 entries)
- A dictionary of job titles (19245 entries)
- A dictionary of honorifics used before names (173 entries)
- A dictionary of country names including variations in spellings (923 entries)
- A dictionary of nick names and laqabs (8169 entries)
- A dictionary of person titles (20 entries)
- a dictionary of words and phrases that act as person indicators such as 'المشرف الرياضي' (The sports supervisor) (421 entries)

5 System Architecture and Implementation

Figure 1 shows the architecture of the PERA system. Our system has two major components: the *gazetteers* and the *Grammar*. A *filtration mechanism* is employed that enables revision capabilities in the system.

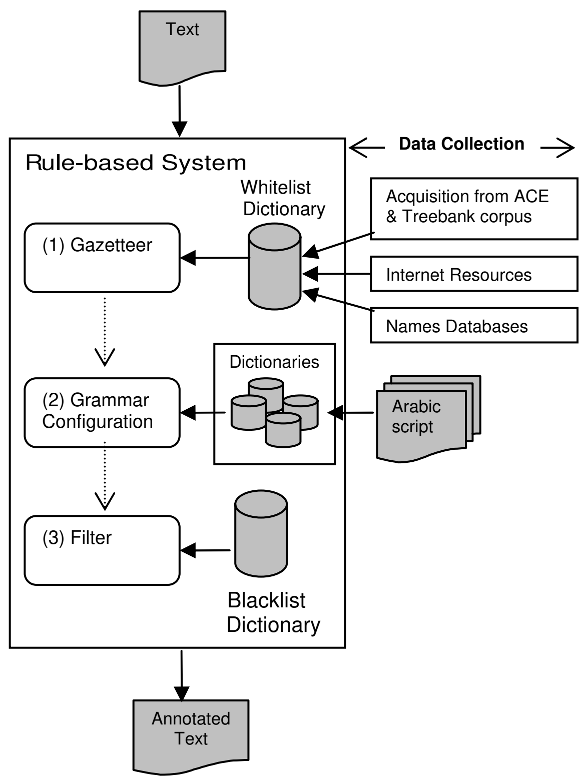

Figure 1: Architecture of the System

5.1 Gazetteers

The main gazetteer (dictionary) of complete person names plays the role of a fixed static dictionary of full person names. It recognizes person name entities by being applied as a *Whitelist* mechanism that accepts matches which are reported as a result of an intersection between the dictionary and the input text. A Whitelist is a list of strings that must be recognized independent of the rules. It contains entries in the following format:

عبدالرحمن قاسم الشيراوى|Abdulrahman Qasim Mohammed Alshirawi

Since the system being developed can be incorporated in various applications independent of language constraints, the English transliterations of the Arabic names are included in the dictionary as meta data.

5.2 Grammar

The grammar performs recognition and extraction of person entities from the input text based on combinations of regular expression patterns. This rule definition is particularly challenging for the Arabic language due to reasons such as:

- Arabic writing systems do not exhibit differences in orthographic case, such as initial capitalized letters to indicate the presence of a proper name. This lack of specific internal structure in the Arabic language poses great challenge for recognizing person entities.
- Arabic is a highly inflected language which entails a requirement of understanding of its morphological nature. The inflected Arabic word maybe composed of prefixes such as prepositions and suffixes such as pronouns. These affixes need to be addressed to ensure recognition of person names alone.

Due to the above complexities in the Arabic language a deep contextual analysis of various Arabic scripts was performed using Python scripts to build grammar rules based on keywords or trigger words forming a window around a person name.

An Example Rule:

The following rule recognizes a person name composed of a first name followed by optional middle and last names based on a preceding person indicator pattern.

```
((honorfic+ws(location(ية|ي)+ws)?)
+firsts_v((ws+middle_vv)|
(ws+lasts_v))?ws+(number)?)
```

Description:

- The names should be verified against their respective dictionaries (i.e. first, middle, and last names).
- The indicator pattern is composed of an honorific such as "الملك" [The king] followed by an optional *Nisba* derived from a location name such as "الأردني" [Jordanian]. These act as trigger words to recognize the person name and should be verified against their respective dictionaries of honorific and locations.
- The rule also matches an optional ordinal number appearing at the end of some names such as "الثاني" [II].
- The Arabic suffix letters "ية" and "ي" used in the above pattern parses the inflections attached to *Nisba* derived from locations that are commonly found in Arabic text.

Implementation:

```
(($honorific$ws*($location
(\x{064A}|\x{0629})*$ws*)?)+
$firsts_v(($ws*$middle_vv)|
($ws*$lasts_v))?$ws*($number)?)
```

Writing conventions:

- `$`: reference to a slave schema.
- `Firsts_v`: dictionary of first names.
- `Middle_vv`: dictionary of middle names.
- `Lasts_v`: dictionary of last names.
- `Ws`: whitespace.
- `Honorific`: dictionary of honorifics appearing before names.
- `Location`: dictionary of locations.
- `Number`: Arabic ordinal numbers.

Example:

The following name would be recognized by the above rule:

الملك الأردني عبد الله الثاني

[The Jordanian king Abdullah II]

Apart from contextual cues, the typical Arabic naming elements were used to formulate rules such as nasab, kunya, etc. Thereby the rules resulted in a good control over critical instances by recognizing complex entities.

5.3 Filter

A *filtration* mechanism is used in the form of a *Blacklist* (rejecter) within the grammar configuration to filter matches that appear after person titles but are invalid person names. In the following example:

‘*وزير الخارجية العراقي الامين العام*’ [The Iraqi Foreign Minister the Secretary-General]

The sequence of words ‘وزير الخارجية العراقي’ [The Iraqi Foreign Minister] acts as a person indicator and the word immediately following it is usually a valid person name. However, in this example, the words following the person indicator that is, ‘ الامين العام’ (the Secretary-General) is not a valid person name. Hence the role of the blacklist comes into play by rejecting the incorrect matches recognized by certain rules.

5.4 The Implementation Platform

The PERA system was implemented through incorporation into the FAST ESP framework, (FAST,_). FAST ESP is an integrated software application that provides searching and filtering services. It is a distributed system that enables information retrieval from any type of information, combining real-time searching, advanced linguistics, and a variety of content access options into a modular, scalable product suite.

The document processing stage within FAST ESP system provides support for Entity Extraction. PERA is implemented through the customizable document processing *pipelines* within FAST ESP, which consists of multiple document processing *stages*. A new search pipeline was created and stages containing the grammar configuration and gazetteers were added to this pipeline. Figure 2 indicates the functionality of the PERA system incorporated in the pipeline within FAST ESP for recognizing and tagging person entity in text.

6 The Experiment

In evaluating the PERA system we follow the standard practice in the IE field of comparing system output against a reference corpus and measuring the performance of the Arabic person named entity.

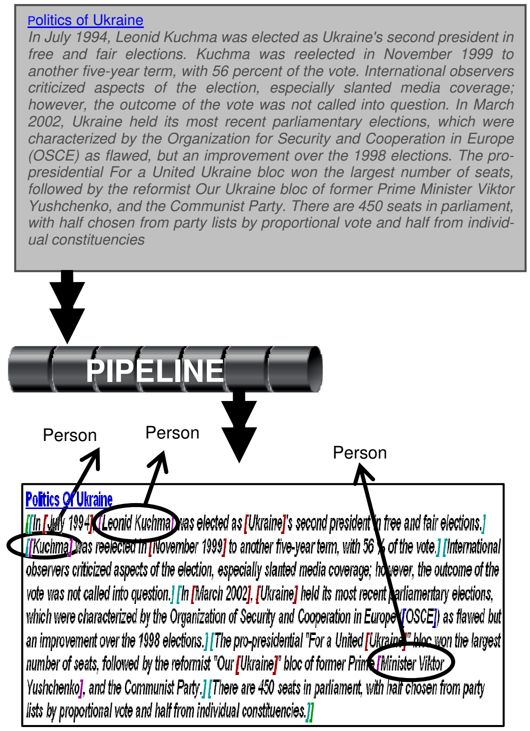

Figure 2: PERA incorporated into FAST ESP pipeline to produce Tagged text

6.1 Reference Corpus

The text within the ACE and Treebank corpus was used for creating the entity tagged reference corpus for evaluating PERA. The text was chosen randomly from files with ‘sgm’ extension (containing the Arabic script) within ACE & Treebank corpus. The tagging was automatically performed with a Python script and further a post manual check was performed to correct any invalid tags or identify the missing ones. The end product was an annotated text corpus in the *xml* format with the *UTF-8* encoding. This was divided into a 46 test sets and each evaluated individually with hurricane. The total size of the reference corpus build is around 4MB. The size and content of the corpus is such that it contains a representative amount of occurrences of the person entity.

6.2 Evaluation Method

We have adopted the evaluation measures that are standard in the IE community (De Sitter et al., 2004), to evaluate and compare the results (precision, recall and F-measures):

$$\textbf{Precision} = \frac{\text{correct entities recognized}}{\text{total entities recognized}}$$

$$\textbf{Recall}= \frac{\text{correct entities recognized}}{\text{total correct entities}}$$

$$\textbf{F-measure}= \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Precision indicates how many of the extracted entities are correct. *Recall* indicates how many of the entities that should have been found, are effectively extracted. Usually there is a trade off of recall against precision. Therefore, often an average accuracy is reported in the form of the *F-measure*, a harmonic mean which weights recall and precision equally. It was introduced to provide a single figure to compare different systems' performances. The PERA system implemented within the FAST ESP pipeline was evaluated using an Information Extraction testing tool called 'hurricane' that applies these standard measures.

6.3 Results

Figure 3 is a snapshot of the evaluation performed by hurricane in terms of the above mentioned measure.

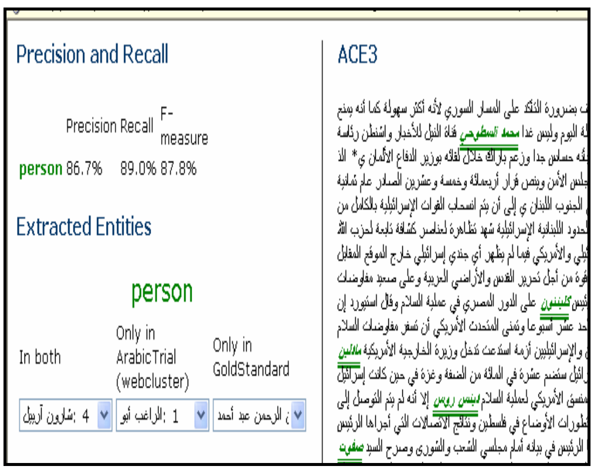

Figure 3: An Extraction from Hurricane Evaluation

The extraction quality of the pipeline created for the person name extractor confirms to the initial target set. The required degree of precision (80%) and recall (70%), for the Person name extractor, has been achieved with the hurricane evaluation. Some of the entries within the gazetteers were extracted from the same corpus used also for creating the reference corpus for evaluation. However, the results achieved are accurate since they indicated recognition of person entities not included in the gazetteers but being recognized by the grammar rules.

Table1 indicates the performance figures produced by 6 out of the 46 sets used for Hurricane evaluation.

The average Precision and Recall for the total 46 sets in recognizing person names is 85.5% and 89%, respectively. And the average f-measure is 87.5%.

Test Set	Precision	Recall	F-measure
Treebank set 1	91.2	90.3	90.7
Treebank set 2	94	96.3	95.1
Treebank set 3	84.2	84.7	84.4
ACE set 1	89.6	96.8	93.1
ACE set 2	88.4	94.2	91.2
ACE set 3	86.7	89	87.8

Table 1: Evaluation result for 6 test sets.

The missing accuracy can be overcome in the following ways:

- Expanding the dictionary of person names further.
- More Arabic text/corpus can be analyzed to identify strings that act as person indicators.
- Reducing negative effects on evaluation results (true positive being treated as false positives) caused due to incomplete annotation of the test corpus. The reference corpus can be further fine tuned to tag all person entities completely.
- Enhancing quality of transliterated names used.
- Using Arabic text with error free spelling.
- Including all possible spelling variations used for names in Arabic written text.

7 Conclusion and Future Work

The work done in this project is an attempt to broaden the coverage for entity extraction by incorporating the Arabic language, thereby paving the path towards enabling search solutions to the Arabian market.

Various data collection techniques were used for acquiring gazetteer name lists. The rule-based approach employed with great linguistic expertise provided a successful implementation of the PERA system. Rules are capable of recognizing inflected forms by breaking them down into stems and affixes. A filtration mechanism is employed in the form of a rejecter within the grammar configuration that helps in deciding where a name ends and the non-name context begins. We have evaluated our system performance using a reference corpus that is tagged in a semi-automated way. The average Precision and Recall achieved for recognizing person names was 85.5% and 89%, respectively. Suggestions for improving the system performance were provided.

This work is part of a new system for Arabic NER. It has several ongoing activities, all concerned with extending our research to recognize and categorize other entity Arabic named entities such as locations, organization.

Acknowledgement

This work is funded by the "Named Entity Recognition for Arabic" joint project between The British Univ. in Duabi, Dubai, UAE and FAST search & Transfer Inc., Oslo, Norway. We thank the FAST team. In particular, we would like to thank Dr. Petra Maier and Dr. Jürgen Oesterle for their technical support.

Any opinions, findings and conclusions or recommendations expressed in this material are the authors, and do not necessarily reflect those of the sponsor.

References

Saleem Abuleil 2004. Extracting Names from Arabic Text for Question-Answering Systems, *In Proceedings of Coupling approaches, coupling media and coupling languages for information retrieval (RIAO 2004)*, Avignon, France. pp. 638- 647.

Da'ud Ibn Auda. 2003. Period Arabic Names and Naming Practices, *In Proceedings of the Known World Heraldic Symposium* (SCA: KWHS Proceedings, 2003), pp. 42-56, St. Louis, USA.

FAST ESP http://www.fastsearch.com/thesolution.aspx?m=376

Nancy Chinchor 1998. Overview of MUC-7. *In Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Available at: http://www.itl.nist.gov/iaui/894.02/related_projects/muc/

Leah S. Larkey, Nasreen Abdul Jaleel, Margaret Connell. 2003. *What's in a Name?: Proper Names in Arabic Cross Language Information Retrieval* CIIR Technical Report IR-278. Available at http://ciir.cs.umass.edu/pubfiles/ir-278.pdf

John Maloney and Michael Niv. 1998. TAGARAB: A Fast, Accurate Arabic Name Recogniser Using High Precision Morphological Analysis. *In Proceedings of the Workshop on Computational Approaches to Semitic Languages.* Montreal, Canada. August, pp. 8-15.

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Irina Temnikova, Anna Widiger, Wajdi Zaghouani, and Jan Zizka. 2005. Multilingual person name recognition and transliteration. Journal CORELA-Cognition, Représentation, Langage, Vol. 2, ISSN 1638-5748. Available at http://edel.univ-poitiers.fr/corela/

Doaa Samy, Antonio Moreno and Jose M. Guirao. 2005. *A Proposal for an Arabic Named Entity Tagger Leveraging a Parallel Corpus*, International Conference RANLP, Borovets, Bulgaria, pp. 459-465.

An De Sitter, Toon Calders, and Walter Daelemans. 2004. A Formal Framework for Evaluation of Information Extraction, University of Antwerp, Dept. of Mathematics and Computer Science, Technical Report, TR 2004-0. Available at http://www.cnts.ua.ac.be/Publications/2004/DCD04

Antonio Toral. 2005. DRAMNERI: a free knowledge based tool to Named Entity Recognition. *In Proceedings of the 1st Free Software Technologies Conference.* A Coruña, Spain. pp. 27-32.

Imed Zitouni, Jeffrey Sorensen, Xiaoqiang Luo and Radu Florian, 2005 The Impact of Morphological Stemming on Arabic Mention Detection and Coreference Resolution, *In the Proceedings of the ACL workshop on Computational Approaches to Semitic Languages,* 43rd Annual Meeting of the Association of Computational Linguistics (ACL05). June, Ann Arbor, Michigan, USA, pp. 63-70.