Adapting a resource-light highly multilingual Named Entity Recognition system to Arabic

Wajdi Zaghouani[a], Bruno Pouliquen[b], Mohamed Ebrahim[b], Ralf Steinberger[b]

[a]Linguistic Data Consortium, University of Pennsylvania
[b]European Commission – Joint Research Centre

E-mail: wajdiz@ldc.upenn.edu, {Bruno.Pouliquen, Mohamed.Ebrahim, Ralf.Steinberger}@ jrc.ec.europa.eu

Abstract

We present a working Arabic information extraction (IE) system that is used to analyze large volumes of news texts every day to extract the named entity (NE) types person, organization, location, date and number, as well as quotations (direct reported speech) by and about people. The Named Entity Recognition (NER) system was not developed for Arabic, but - instead - a highly multilingual, almost language-independent NER system was adapted to also cover Arabic. The Semitic language Arabic substantially differs from the Indo-European and Finno-Ugric languages currently covered. This paper thus describes what Arabic language-specific resources had to be developed and what changes needed to be made to the otherwise language-independent rule set in order to be applicable to the Arabic language. The achieved evaluation results are generally satisfactory, but could be improved for certain entity types.

1. Introduction

News aggregation services such as *Google News* and *Yahoo! News* scrape tens of thousands of media web sites per language, cluster them and allow users search word-based filtering to identify news items of potential interest. As in any other Information Retrieval (IR) setting, users need to look through the whole set of documents to find the information they need. News analysis systems like *NewsVine*[1], *SiloBreaker*[2] and the *Europe Media Monitor* (EMM) application *NewsExplorer*[3] (Steinberger et al. 2009) go beyond this IR setting, by additionally extracting information from the news text and by further linking information found in the news.

Most such systems are monolingual (typically English). NewsExplorer, in contrast, currently covers 19 languages, including Arabic. Such high multilinguality is most likely to be achieved only if the effort for each language is limited. The EMM family of applications process an average of about 100,000 articles per day. The world language Arabic is relatively well represented within EMM.

Steinberger et al. (2008) propose to use language-independent rules and as few language-specific resources as possible. These should furthermore be simple and easy to produce, and they should be organized in a compositional manner so that any new language can simply be 'plugged in' to the overall system once the language-specific resources are available. In this paper, we present the effort of adding the Semitic language Arabic to EMM-NewsExplorer. Arabic is significantly different from the other, mostly Indo-European EMM languages (Vergyri et al. 2004), and is thus a good test case for the proposed method. For NER, the major differences are that Arabic does not distinguish upper and lower case letters (uppercase helps to identify the beginning and end of potential NEs), that it uses both prefixes and post-fixes, and that short vowels are frequently not written (Debili & Achour 1998).

In the next section we discuss the state of the art of Arabic NER. Section 3 illustrates the architecture of the system and its various components. In Section 4, we present the evaluation results achieved with the described method. Section 5 summarizes the results.

2. Related work

NER by now is a known and common task. For a few widely-used languages, a large variety of NER tools exist (Nadeau & Sekine 2009). A remaining challenge in the field is how to develop such systems quickly for resource-poor languages.

For Arabic, available software is mostly commercial, including *IdentiFinder*[4] (BBN), *NetOwlExtractor*[5] (NetOwl), *Siraj*[6](Sakhr), *Clear Tags*[7] (ClearForest) and *InXight-Smart-Discovery-Entity-Extractor*[8] (InXight). Little information is available on the inner working and on formal evaluation results of these systems.

[1] http://www.NewsVine.com (all the mentioned web sites were last visited in March 2010).
[2] http://www.SiloBreaker.com
[3] http://press.jrc.it/overview.html
[4] http://www.bbn.com/technology/speech/identifinder
[5] http://www.sra.com/netowl/entity-extraction/
[6] Online demo version available at: http://siraj.sakhr.com/
[7] http://www.clearforest.com/solutions.html
[8] http://www.inxightfedsys.com/products/sdks/tf/default.asp

Benajiba et al. (2007) developed *ANERsys*, a NER system based on a Maximum Entropy-based statistical learning model. Benajiba additionally uses a manually created dictionary to boost the system performance. Benajiba tested the system on a purpose-built corpus and obtained a precision of 63.21%, a recall of 49.04% and an F-measure of 55.23% with four categories (location, person, organization and a miscellaneous entity category).

Maloney and Niv (1998) presented TAGARAB, an Arabic name recognizer that combines the pattern matching module with a morphological analyzer to improve performance. The overall performance obtained by TAGARAB for the various categories (time, person, location and number) was a precision of 89.5 %, a recall of 80.8 % and an F-measure of 85 %.

Abuleil (2004) developed a rule-based system that makes use of hand-written rules and trigger words. Abuleil's system obtained a precision of 90 % on people, 93 % on location and 92 % on organization.

Shaalan and Raza (2009) presented a NER system for Arabic (NERA) using a rule-based approach, dictionaries and a local grammar. NERA obtained an F-measure of 87.7% for people, 85.9% for locations, and 83.15% for organizations.

Traboulsi (2009), finally, discussed the use of local grammars to build an Arabic NER system.

Similarly, our own system is also based on hand-written local patterns. However, a big difference is that we use a set of language-independent rules in combination with language-specific parameter files, containing the relevant vocabulary and – possibly – extra language-specific rules. The general mechanism will be explained in Section 3.3. In the same section, we will also show what needed to be done in order to deal with Arabic-specific differences.

3. The proposed Arabic NER system

We will first present the architecture of the proposed system. Then we will describe the morphological pre-processing step and the working of the NER rules.

3.1. The Architecture of the system

The architecture of the system is shown in Figure 1. The system relies on 3 main processing steps: Pre-processing (segmentation rules), lookup of full known names, and recognition of unknown names using local grammars and a set of dictionaries.

Names found repeatedly in the course of long-term multilingual news analysis are stored in a database. The database currently contains over one million known names, plus hundreds of thousands of known variants for these names. Every day, lists of known entities are exported to a finite-state automaton, which identifies the known names in the news texts before the extraction rules

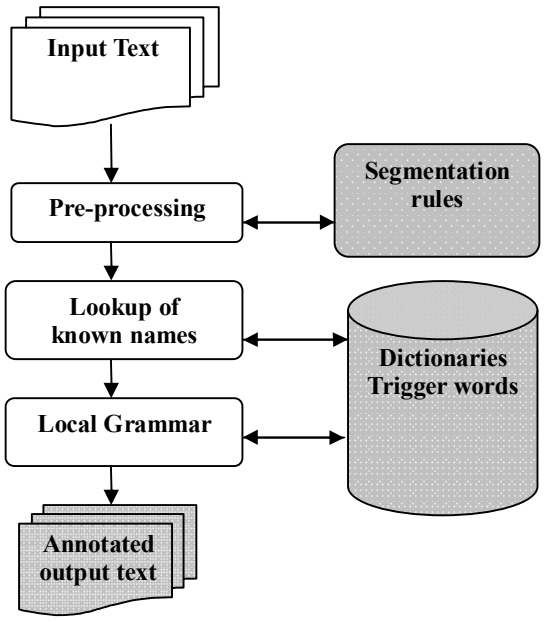

Figure 1: Architecture of the System.

apply. Every now and then, we also search known names and their variants on Wikipedia and – if found – we add the multilingual name variants found there to the list of known entities (Pouliquen et al. 2005).

3.2. Morphological pre-processing

Arabic morphology is relatively complex in that it uses prefixes, infixes and suffixes, not only for inflection but also to concatenate words. This various morphological variation can be dealt with by using hand-crafted rules, which enable to strip off possible prefixes and suffixes from the word stem before applying the NER grammars (Shaalan and Raza 2008). For each type of NE, several rules were built and each one is applied to all input words. By doing this, particles attached to words are stripped, allowing a better match between the words found in the text and those in the dictionaries. For example, the attached conjunction ***wa***, the preposition ***li*** and the definite article ***al*** are stripped off all words and names, like in the example below. Transformation 1 thus consists of removing the attached conjunction و wa (and), and transformation 2 removes the attached preposition ل /li/ (for) and the definite article ال / il / (the):

وللرئيس /**walil**ra'iis /	(and for the president).
للرئيس /**lil**ra'iis/	(for the president).
رئيس /ra'iis/	(president).

3.3 Local Grammars for person and organization name recognition

We will first explain how the multilingual EMM grammars for NER work in general. We will then describe the Arabic-specific differences.

The rule set is mostly *language-independent* (meaning that the same rules are applied to all languages), but they make reference to *language-specific* words, multi-word expressions and regular expressions that are stored in language-specific parameter files. Due to this modularity, it is relatively easy to add a new language to the tool set. Whenever a rule needs to be added for a specific language, as it is the case for Arabic, this rule will be stored in the language-specific parameter file. The language-specific resources are hand-compiled, but bootstrapping methods are used to build the word lists quickly (Pouliquen et al. 2005). We refer to the language-specific words and expressions that are part of the local patterns and that help our system to identify named entities as *trigger words*. We use this unspecific term because not only titles are included, but also verbal phrases, regular expressions, and more. The trigger word lists include titles (السيدة *Mrs.,* استاذ *Prof.,* دكتور *Dr.*, etc.), professions or positions (مدير *Director,* رئيس *President,* محام *lawyer,* الكاهن *priest*, etc.), country adjectives (التونسي *Tunisian,* الكندي *Canadian*, etc.), religious and ethnic groups (الكاثوليكي *Catholic,* السنية *Sunni,* بربر *Berber*) and many other expressions indicating that some uppercase words (X) may be names (e.g. '*X declared*', '*X died*', *[0-9]+-year-old X*, etc.), and more. The word lists for each language are kept in a language-specific parameter file. The language-specific parameter file furthermore contains a word list loosely named *modifiers*, i.e. words that can appear in certain places between the name mention and the trigger words. The *modifier* list can contain all sorts of modifiers, but also auxiliary verbs and more (e.g. 'has', 'yesterday'). The rule set thus makes reference to different subsets of these language-specific word lists. Here are some examples, using the notation: *(\w+)* represents an unknown word, *\b* the obligatory whitespace; + indicates one or more elements, * means zero or more elements:

(1) PERSON_TRIGGER+\bUppercaseWord\b UppercaseWord
(2) UppercaseWord\bUppercaseWord(\bMODIFIER)*\b PERSON_TRIGGER+

Rule (1) simply says: recognize any combination of at least two uppercase words as person names if they are found next to trigger words or expressions (e.g. *Mr. Ahmed Issa*). Note that we require at least two name parts because all names are unambiguously grounded to real-world entities so that the extracted information can be displayed on NewsExplorer.[9] Rule (2) captures apposition constructions such as *Hamid Karzai, the newly elected Afghan president*. The words *the, newly* and *elected* can be found in the English modifier lists, while both *Afghan* and *president* are part of the trigger word lists. Our recognition expressions (e.g. for modifier) can themselves be more complex than what is mentioned here. Details of the approach that are worth highlighting here are: (a) Determiners, adverbs and other elements may be loosely combined under the heading *modifier*, as the application would not benefit from a distinction; (b) the order of elements within the groups *modifier* and *person trigger* is also not specified. These are both examples of *under-specified* rules, that make it easier to write the rules and to apply it to different languages. While *generation* grammars would need this information, our *recognition* grammars do not.

These generic rules make reference to case information, which is not available in Arabic or Farsi. When a person trigger expression is found in these Arabic script languages, we cannot know whether the preceding or following words are names or not (because we do not have access to generic dictionaries, part-of-speech or syntactic information). Furthermore, we would not know where the name borders are. For this reason, we needed to write separate, safer rules, which were then placed in the language-specific parameter file so that they only apply to Arabic. The following are examples of such Arabic-specific rules:

(3) KNOWN_NAME+\b(\w+)\bNAME_INFIX*\b KNOWN_NAME
(4) (\w+)\bNAME_INFIX+\b(\w+)
(5) PERSON_TRIGGER+\b(\w+)\bKNOWN_NAME
(6) NAME_STOP_WORDS\b(\w+)(\bMODIFIER)*\b PERSON_TRIGGER+

Rule (3) recognizes combinations of known name parts (first names or last names – in some Arabic countries, the distinction is mostly irrelevant). The names can optionally be separated by one or more name parts (e.g. بن *bin,* عبد *abd,* أبو *abu,* آل *Al*), referred to as *name infixes*, to stay in line with other languages where we can find name infixes such as *van der, de la, della, von*, etc. This rule would successfully recognize the name محمد علي بن حليمة (*Mohammed ali ben Halima)*, assuming that both *Mohammed* and *Halima* are part of the list of known names. The known name list used contains thousands of name parts from different parts of the Arab world. Similar lists exist for most EMM languages. Rule (4) recognizes combinations of unknown (or – optionally – known) words if they are linked by name infixes. Similar rules will also allow for longer combinations of three, four or more names, as long as each element is linked to the others via name infixes. Rule (5) will recognize an unknown word and a known name part as a name if they follow a trigger expression (e.g. السيد عيسى أحمد – *Mr. Issa Ahmed*, assuming that *Ahmed* is a known name and *Issa* is unknown). Rule (6) is the Arabic equivalent to rule (2), i.e. it will capture apposition constructions. However, in order to recognize the left-hand-side border of the name, we make use of *name stop words*. Name stop words are typically high-frequency words from a long list of words that can never be name parts. In the example below, the rule will thus recognize *Hamid Karzai* or any other name as a person, even if the words involved are not known name parts. The words *and said*, found in the *name stop*

[9] See, e.g., the page for Iraqi politician *Nouri al-Maliki*: http://emm.newsexplorer.eu/NewsExplorer/entities/en/77049.html

word list, ensure that the left-hand-side border of the name is correctly identified as *Hamid*. The uppercase condition applicable to most EMM languages is thus replaced in Arabic by the introduction of the name stop word list:

وقال حامد كرزاي الرئيس الأفغاني المنتخب الجديد
And said Hamid Karzai, the newly elected Afghani president

An alternative would have been to use a part-of-speech tagger or full dictionaries with part-of-speech information, but developing these would have been a lot of effort. More importantly, the parallelism between languages would have gone lost. While new rules needed to be introduced especially to deal with the Arabic language, the format of the Arabic language-specific resources remains the same.

3.4 Resources required

The rules thus make reference to words and multi-word expressions that are stored in various dictionary files. The person name recognition tool distinguishes various types of trigger words, lists of modifiers and of name stop words. These trigger word lists have been created manually, but using semi-automatic procedures, by looking at the most frequent left and right-hand-side contexts of known Arabic NEs, or at the context of NEs that are found using the rules with initial lists of seed words. In addition, the resource contains an extensive list of common name parts (not only Arabic first names but also common international names like جون John, جان Jean, خوان Juan written in Arabic) and a list of *name infixes* (بن *bin,* عبد *abd,* أبو *abu,* آل *Al, etc.*). The name part lists contain altogether 19.600, names, collected from various Arabic Internet resources, including the Arabic version of Wikipedia.

Locations are recognized through a simple gazetteer lookup procedure, without any grammatical patterns. The gazetteer consists of currently 2,200 names of countries, cities, towns and villages found mainly in the multilingual KNAB gazetteer produced by the *Institute of Estonian Language*[10].

Organization names can be rather complex, so that our organization recognition tool is limited to a simple lookup of 4.000 well-known (mostly Arabic) company and organization names.

3.5 Difficulties and challenges

At times, it was difficult to create rules because of the complexity of organization and person names in Arabic. A major challenge was to predict the boundaries of the named entities, especially with long and composed Arabic names. We found cases where the full name is composed of 8 words.

[10] http://www.eki.ee/knab/p_mm_en.htm

Moreover, it was very difficult to cover with our local grammar all regional variants of the names of organizations because labels and standards differ from one country to another and from one culture to another. For instance, companies that originate from the Maghreb region will frequently use French words in their names (like محمد داوود بياس أوتو – *Mohamed Daoud Pièces Auto*). On the other hand companies in the Gulf region will use English words (such as مدني تايلورز – Madani Tailors). We have therefore limited our coverage to a lookup of major internationally known organizations, as well as those from the Arab world.

4. Evaluation

4.1 Corpus

The evaluation corpus was built from online news sources covering in equal parts the Tunisian newspaper Assabah[11] and the Lebanese newspaper Alanwar[12]. Our corpus consists of 35 news articles and 34.000 tokens (17.316 for Assabah and 16.684 for Alanwar). We have manually tagged the NEs in the corpus. Table 1 shows the NEs manually tagged in this corpus, and their distribution ratio.

Category	Distribution	Ratio
Person	804	43.43%
Location	433	23.39%
Organizations	514	27.76%
Dates	54	2.91%
Numeric expression	46	2.48%
Total	**1851**	**100%**

Table 1: Distribution and ratio of Named Entities in the evaluation corpus.

4.2 Results

Table 2 summarizes the results obtained by our system against our evaluation corpus. We have included Precision, Recall and F-measure values.[13]

Category	Precision	Recall	F-measure
Person	87 %	66.54 %	75.40 %
Organization	69.96 %	35.79	47.35 %
Location	91.52 %	74.82 %	82.33 %
Date and time	96.13 %	94.11 %	95.10 %
Numeric expression	93.29 %	89.47 %	91.34
Overall	**87,17 %**	**65,74 %**	**74,95 %**

Table 2: Results obtained for the various Named Entity types.

[11] http://www.assabah.com.tn/
[12] http://www.alanwar.com/ar/
[13] These results were produced *not* including rule (6), from Section 3.3.

4.3 Error Analysis

An analysis of the detection errors has revealed many cases of wrong categorization due to the high ambiguity of some Arabic words. Another type of error was that names were only partially recognized. On the other hand, the absence of rigorous standards of writing Arabic text has lead to inconsistencies in the spelling of some words and therefore has influenced our results. For example the writing of the letter *Hamza* ء, an Arabic glottal stop, is often omitted at the beginning of words (e.g. a name like *Ahmad* could be written as احمد or as أحمد).

5. Conclusion

We have presented work on adapting a multilingual NER system to the Arabic language. The otherwise mostly language-independent rules had to be adapted to Arabic, mostly because the lack of case information makes it difficult to know where a name starts and ends. More than for other languages, we needed long lists of potential name parts, and we had to make much more use of name stop words. Not having access to full dictionaries and to part-of-speech information made the NER task rather difficult for Arabic, more so than for the EMM languages using the Latin script (and thus distinguishing case information). In NewsExplorer, we are currently mostly making use of *safe* NER rules (such as those using lists of known name parts and name infixes). The reason is that we did not have an Arabic speaker in the group for a long time. Not having any control instance, we needed to optimize precision, at the cost of lowering recall. However, we are now planning to work on improving the recall. Having said this, we feel that the results are relatively good, considering the simplicity of the approach and having ensured that the approach remains the same across all 20 languages in which we currently recognize named entities.

6. References

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

Benajiba Y., Rosso P. and Benedi J. M. (2007). ANERsys: An Arabic Named Entity Recognition System based on Maximum Entropy. In *Proceedings of the 2007 Conference on Computational Linguistics and Intelligent Text Processing*. Springer-Verlag, LNCS(4394), pp. 530-541. Mexico City, Mexico.

Debili, F. and Achour H. (1998). Voyellation automatique de l'arabe. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*. Montreal, Canada, pp. 42-49.

Maloney, J. and Niv. M. (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, pp. 8-15.

Nadeau D. & S. Sekine S. (2009). A survey of named entity recognition and classification. In: S. Sekine & E. Ranchhod (eds.), *Named Entities – Recognition, Classification and Use*. Benjamins Current Topics, Volume 19. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Pouliquen B., Steinberger R., Ignat C., Temnikova I., Widiger A., Zaghouani, W. & Žižka J. (2005). Multilingual person name recognition and transliteration. *Corela, Numéros spéciaux, Le traitement lexicographique des noms propres.*

Shaalan K. & Raza H. (2008). Arabic Named Entity Recognition from Diverse Text Types. In Proceedings of the 6th International Conference GoTAL. Gothenburg, Sweden, pp. 440-451.

Shaalan, K. & H. Raza (2009). NERA: Named Entity Recognition for Arabic. *The Journal of the American Society for Information Science and Technology (JASIST)*. NJ, USA, John Wiley & Sons, Inc. 60(8): pp. 1652-1663.

Steinberger R., Pouliquen B. & Van der Goot E.. (2009). An Introduction to the Europe Media Monitor Family of Applications. In Fredric Gey, Noriko Kando & Jussi Karlgren (Eds.): *Information Access in a Multilingual World - Proceedings of the SIGIR 2009 Workshop (SIGIR-CLIR'2009)*. Boston, USA, pp. 1-8.

Steinberger R., Pouliquen B. and Ignat C.. (2008). Using language-independent rules to achieve high multilinguality in Text Mining. In Fogelman-Soulié Françoise, Domenico Perrotta, Jakub Piskorski & Ralf Steinberger (Eds.), *Mining Massive Data Sets for Security*. Amsterdam, The Netherlands, IOS Press, pp. 217-240.

Traboulsi. H. (2009). Arabic Named Entity Extraction: A Local Grammar-Based Approach. In *IMCSIT'2009*, Mrągowo, Poland, pp. 139-143.

Vergyri, D., Kirchhoff, K., Duh, K. et Stolcke A. (2004). Morphology-Based Language Modeling for Arabic Speech Recognition. In *International Conference on Spoken Language Processing (ICSLP)*. Jeju Island, Korea, pp. 2245-2248.