NERA: Named Entity Recognition for Arabic

Khaled Shaalan and Hafsa Raza
Faculty of Informatics, The British University in Dubai, P.O. Box 502216, Dubai, United Arab Emirates.
E-mail: khaled.shaalan@buid.ac.ae; hafsa.raza@gmail.com

Name identification has been worked on quite intensively for the past few years, and has been incorporated into several products revolving around natural language processing tasks. Many researchers have attacked the name identification problem in a variety of languages, but only a few limited research efforts have focused on named entity recognition for Arabic script. This is due to the lack of resources for Arabic named entities and the limited amount of progress made in Arabic natural language processing in general. In this article, we present the results of our attempt at the recognition and extraction of the 10 most important categories of named entities in Arabic script: the person name, location, company, date, time, price, measurement, phone number, ISBN, and file name. We developed the system *Named Entity Recognition for Arabic* (NERA) using a rule-based approach. The resources created are: a Whitelist representing a dictionary of names, and a grammar, in the form of regular expressions, which are responsible for recognizing the named entities. A filtration mechanism is used that serves two different purposes: (a) revision of the results from a named entity extractor by using metadata, in terms of a Blacklist or rejecter, about ill-formed named entities and (b) disambiguation of identical or overlapping textual matches returned by different name entity extractors to get the correct choice. In NERA, we addressed major challenges posed by NER in the Arabic language arising due to the complexity of the language, peculiarities in the Arabic orthographic system, nonstandardization of the written text, ambiguity, and lack of resources. NERA has been effectively evaluated using our own tagged corpus; it achieved satisfactory results in terms of precision, recall, and F-measure.

Introduction

A *Named Entity Recognition* (NER) system is a significant tool in natural language processing (NLP) research since it allows identification of proper nouns in open-domain (i.e., unstructured) text. For the most part, such a system is simply recognizing instances of linguistic patterns and collating them. Larkey, Abdul Jaleel, and Connell (2003) conducted a study that showed the importance of the proper names component in language tasks involving searching, tracking, retrieving, or extracting information. Another study by Crestan and de Loupy (2004) showed that named entity extraction helps users to more quickly and efficiently browse large document collections. This seems plausible because according to Gey (2000), 30% of the content-bearing words in news are proper names. Abuleil (2004) and Chinchor (1998) stated that the valuable information in text is usually located around proper names, so identifying proper names is an important first step.

In the 1990s, the NER concept was introduced at the Message Understanding Conferences (MUCs), which were financed by the Defense Advanced Research Projects Agency to encourage the development of new and better methods of information extraction. At the sixth conference (MUC-6; http://cs.nyu.edu/cs/faculty/grishman/muc6.html) the task of named entity recognition was defined as three subtasks: ENAMEX (for the person, location, and organization names), TIMEX (for date and time expressions), and NUMEX (for monetary amounts and percentages). Until now, NER systems developed in various languages have evolved around these three subtasks; however, we have broadened the coverage of the named entities with our system NERA, which identifies 10 types of phenomena, including Person names, locations, companies, dates, time, prices, measurements, phone numbers, ISBNs, and file names.

The work presented in this article concentrates on the role of NER in an information-extraction task that retrieves relevant information from a large amount of diverse data. We have adopted the rule-based approach using linguistic grammar-based techniques to develop NERA. The approach is motivated by the characteristics and peculiarities of the Arabic language. The recognition process requires two cycles (Shaalan & Raza 2007, 2008): (a) using the Whitelist component for matching relatively simple linguistic items such as person names and (b) applying the grammar rules involving relatively complex linguistic structures such as NE indicators. The set of grammar rules was derived by analyzing the local lexical context of a large amount of diverse data. A complementary process, which uses metadata (Blacklist or rejecter) about ill-formed NEs, is applied to filter recognition results

Received October 18, 2008; revised March 12, 2009; accepted March 12, 2009

 DOI: 10.1002/asi.21090

TABLE 1. Examples of inflections in Arabic text.

Arabic example	English translation	Entity type	Affix (clitics)
والإمارات العربية المتحدة	and the United Arab Emirates	Location	'و' (Waw)
لباكستان	to Pakistan	Location	'ل' (laam)
بالولايات المتحدة	for the United States	Location	'بال' (baa, alif-laam)
بشبكة اي.بي.سي للتلفزيون الاميركية	by ABC Network for the American television	Company	'ب' (baa)
لهيئة الاذاعة البريطانية " بي بي سي"	for the British Broadcasting Corporation "BBC"	Company	'ل' (laam)
الـ 2925 متر	the 2925 meter	Measurement	'الـ' (alif-al)
لثلاث سنوات	for 3 years	Measurement	'ل' (laam)
بـ 20.266 دولارا	for $20,266	Price	'ب' (baa)

to discard incorrect matches. Sometimes identical or overlapping textual matches are inevitable, resulting in ambiguous NEs. In this case, a heuristic disambiguation technique is applied to get the correct choice with respect to the context in which an ambiguous situation arises. This open-architecture approach provides flexibility and adaptability features in our system so that it can be easily configured to work with different languages, NLP applications, and domains. The NERA system has been evaluated using a reference corpus that is tagged with names in a semi-automated way. The system performance results achieved were satisfactory when evaluated against the standard measures: precision, recall, and F-measure.

The rest of this article is structured as follows. We first highlight how NERA provides solutions to challenges posed by the Arabic language and then present previous related work in Arabic NER. Next, the data-collection methods are described. The following section explains in detail our approach to NER in terms of system architecture. Then, we briefly present an idea about the implementation platform. The subsequent section is dedicated to describing the reference corpora we built to carry out our experimental work. We present the results of our experiments, and then draw some conclusions and discuss future work.

Challenges Tackled by NERA

In NERA, we addressed major challenges posed by NER in the Arabic language arising due to the complexity of the morphological system, peculiarities in the Arabic orthographic system, nonstandardization of the written text, ambiguity, and lack of resources. The following subsections discuss these issues and how we deal with them in NERA.

Complex Morphological System

Arabic has as a rich and complex morphological system due to its highly inflected nature (Shaalan, 2005). Any given Arabic lemma has usually more than one word form to represent it, which includes a root, its internal structure, prefixes, suffixes, and clitics. Since we deal with real, published Arabic text that has not been preprocessed in any way, NEs appear in their real context; one important issue, in this respect, is that NEs as other nouns in Arabic may appear preceded by clitics. These clitics may be a conjunction "و" (Waw, and), a preposition "ل" (Laam, for), "ب" (baa, with), or both "ول" (Waw-Laam, and-for), and so on. The internal structure itself includes short vowels and vocalic length, which together carry the bulk of the morphological and morphosyntactic structures, and a consonantal skeleton, which bears the weight of the lexical (semantic) structure. This concatenative strategy to form words in Arabic causes data sparseness; hence, this peculiarity of the Arabic language poses a great challenge to NER systems.

These inflected forms should not be recognized as a part of the extracted NE. To handle this issue in NERA, we use a heuristic method within the pattern-matching engine, which takes into consideration affixes of words within the pattern being processed. Consequently, within the handcrafted rules, we had to expand the possibilities of matching by indicating that the string might be preceded by one or more preclitics that should be stripped from the recognized NEs in the final output. This method performs morphological analysis before recognizing the NE. Morphological analysis is necessary to look into the affixes and see whether a word is an NE. The rules recognize the inflected NE forms by breaking them down into stems and affixes. Since rules were written using a real-data context, the accuracy achieved is quite authentic. Table 1 shows some inflected NE examples that have been dealt with in NERA's grammar for the respective entity type.

Peculiarities in the Arabic Orthographic System

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 in identifying proper nouns. Thus, the problem of identifying proper names is particularly difficult for Arabic because we cannot recognize them in the text by looking at the first letter of the word.

DOI: 10.1002/asi

TABLE 2. Examples of variations in Arabic text.

Arabic example	English translation	Entity type
أندونيسية / أندونيسيا	Indonesia	Location
لوس انجيلس / لوس انجلوس / لوس انجليس / لوس انجيليس	Los Angeles	Location
جوهانسوبورغ/ جوهانسبرغ / جوهانسبورغ / جوهانسورغ	Johannesburg	Location
جلدر / جيلدر / غلدر / غيلدر	Guilder	Price (currency)
رقم الموبيل: ٥٧٥٦٤٥٣ ,الجوال: ٥٧٥٦٤٥٣	Mobile no. 3546575	Phone no.

TABLE 3. Examples of typographic variations in Arabic text.

Arabic example	English translation	Entity type	Typographic variation
أستراليا / استراليا	Australia	Location	Drop of hamza initially, medially, or finally
السعوديه / السعودية	Saudi Arabia	Location	Two dots removed from taa marbouta
آسيا / اسيا	Asia	Location	Drop of the letter madda from the aleph
دولار أميركي / دولار اميركي	American dollar	Price (currency)	Drop of hamza initially, medially, or finally
ليرة / ليره	lira	Price (currency)	Two dots inserted on final haa
فرنك سويسري / فرنك سويسرى	Swiss franc	Price (currency)	Two dots removed from yaa
الاربع / الاربع	4th	Date (day)	Hamza insertion below vs. above aleph

Hence, to tag proper names in Arabic text, we used keywords or indicator words to guide us to the place where one could find them in the text. By using keywords, we marked name phrases that might contain a certain name, then we processed these phrases to extract names. The method adopted in NERA to analyze these phrases and extract the names was the derivation of a set of heuristic rules and their application to parse the phrases to extract the name entities. Some examples of keywords used for identifying the names are:

- Personal names (title): **Mr.** John Adams → **السيّد** جون آدامز
- Personal names (job title): **President** John Adams → **الرئيس** جون آدامز

Nonstandardization of the Arabic Written Text

Arabic text includes many translated and transliterated NEs. Spelling of translated and transliterated proper names in general tends to be inconsistent in Arabic text. Table 2 shows some examples of the inconsistency, although some can be considered as typographical errors.

The extractor can handle, to some extent, the aforementioned spelling variants. Such issues were dealt with within the context-sensitive rules and dictionary-building rules for the NERA system.

Additionally, the extractor is capable of recognizing variations in written Arabic text for the various named entities being recognized. Table 3 contains some example NEs indicating typographic variations.

Ambiguity

The loss of the internal diacritics (e.g., short vowels or shadda) leads to different types of ambiguity in Arabic texts (both structural and lexical) because different diacritics represent different meanings. These ambiguities can be resolved only by contextual information and an adequate knowledge of the language. Apart from ambiguity due to missing diacritics, Arabic—like many other languages—faces the problem of ambiguity between two or more named entities. The following example indicates an ambiguous situation in Arabic script:

احمد اباد لديه اهتمام بالغ بالفلسفة (**Ahmed Abad** has a keen interest in philosophy.)

In the previous example, the boldface text fragment, "احمد اباد" (Ahmed Abad), represents both a person name and a location, thereby giving rise to an ambiguous situation. These situations can be handled in NERA by specifying a filter rule that gives preference on one extractor over the other. Table 4 shows some of the ambiguous situations that the system can handle.

Lack of Resources

We carried out research on the Arabic language NLP tools and resources in general (e.g., corpora, gazetteers, POS taggers, etc.). This led us to conclude that in comparison with other languages, Arabic lacks mature linguistic resources,

DOI: 10.1002/asi

TABLE 4. Ambiguous examples.

Ambiguous example	English translation	Incorrect	Correct
فرنك سويسري 1.6985	1.6985 Swiss francs	Person	Price
15 رمضان الكريم 2005	15th of Ramadan Al karim 2005	Person	Date
جاسم المتحدة للعقارات والصيانة العامه	Jassim united for real estate and general maintenance	Person	Company
1.5 بليون دولار **سنغافورة**	1.5 billion Singapore dollars	Location	Price
شركة أرامكو السعودية	Saudi Aramco	Location	Company
راشيل فيكتوريا كيون	Racheal Victoria Queen	Location	Person
اليزابيث الثانية في مساءا	In the evening Elizabeth II	Time	Person
نقطة تحول في سبتمبر سنة 1954 ... قدم مارتن...	. . . a turning point in September 1954 Martin presented . . .	Measurement	Date

especially free resources available for research purposes. These resources are often limited in both capability and coverage. Thus, efforts were required in building up resources: evaluation corpora and Whitelist dictionaries of NEs and, as a preprocessing task, to build NERA.

As mentioned earlier, the nonstandardization of written Arabic text causes further bottlenecks; the lack of control over written forms of Arabic script leads to the unstructured nature of Arabic text, thereby making Arabic NLP research far more challenging as compared to other languages.

Related Work

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 research efforts 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. Next, we present some of the successful systems that have been produced in this endeavor.

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 nonname context begins. The decision depends on the part of speech of the Arabic word and/or its inflections. For this test set, 14 texts from the AI-Hayat CD-ROM were selected randomly. In addition to manually tagging them, the authors also ran TAGARAB over these 14 texts and used a standard MUC-style scoring program to compare the morphological output of TAGARAB with the "answers" in the hand-tagged version. The evaluation corpus contains 3,214 tokens, of which 2,324 are Arabic words; 1,879 of the latter received morphological features when hand-tagged. The performance achieved for precision, recall, and F-measure for Person NE recognition was 86.2, 76.2, and 80.9%, respectively; for Location NE: 94.5, 85.3, and 89.7%, respectively; for Number NE: 97.7, 97, and 97.3%, respectively; and for Time NE: 91, 80.7, and 85.4%, respectively.

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: (a) marking the phrases that might include names; (b) building up graphs to represent the words in these phrases and the relationships between them; and (c) applying rules to generate the names, classify each of them, and save them in a database. The module has been tested on 500 articles from the Al-Raya newspaper, published in Qatar. In total, it has identified 335 names, missed 92 names, and extracted 8 names mistakenly. The NER accuracy was calculated in terms of precision by the author: People (90.4%), Location (93%), and Organization (92.3%).

Samy, Moreno, and Guirao (2005) used parallel corpora in Spanish and in Arabic, and an NE tagger in Spanish to tag the names in the Arabic corpus. For each sentence pair aligned together, they used 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. The size of the subcorpus used for the experiment is not large (1,200 sentence pairs), but due to its nature and its source, it contains a considerable number of NEs. From the 1,200 pairs of sentences, 300 sentences from the Spanish corpus were selected randomly with their equivalent Arabic sentences. For each pair, the output of the NE tagger was compared to the manually annotated gold-standard set. They have improved the precision by applying a filter to the Arabic words, which omitted the stop words from the possible transliterated candidates. While they reported high precision (i.e., 84% improved to 90%) and recall (97.5%), note that their approach is applicable only when a parallel corpus is available.

Zitouni, Sorensen, Luo, and Florian (2005) 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 (e.g., the president), or pronominal (e.g., she, it). An entity is the aggregate of all the mentions (of any level) that refer to one conceptual entity. This extended definition of the entity has convinced us of the suitability of the approach. The system was trained and evaluated on the

DOI: 10.1002/asi

Arabic Automatic Content Extraction (ACE) 2003 and part of the 2004 data. The test dataset consists of 178 documents from three sources: 38 Arabic Treebank (ATB) documents, 76 broadcast (bnews) documents, and 64 newswire (nwire) documents. The objective of the evaluation was to investigate the usefulness of stem n-gram features in the mention detection system. The stemming n-gram features gave interesting improvement in terms of precision (64.2 vs. 64.4%), recall (55.3 vs. 55.7%), and F-measure (59.4 vs. 59.7%).

A very recent work by Benajiba and Rosso (2008) also experimented with the statistical approach towards NER (person, location, and organization) using probabilistic models; maximum entropy and then further conditional random (CRF) fields. The authors used their own corpus, called ANERcorp, to train and test the CRF model. ANERcorp is composed of a training corpus and a test corpus annotated especially for the NER task. The overall performance combining all features in terms of precision, recall, and F-measure was 86.9, 72.77, and 79.21%, respectively. The results obtained an accuracy improvement by more than 10 points as compared to the entropy model. In a later work (Benajiba, Diab, & Rosso, 2008), the authors reported that ANERsys is subject to further comparative study between many probabilistic models (e.g., SVM, HMM, Maximum Entropy, CRF, etc.) and also experiments using a combination of different models.

Data Collection

Various methods and techniques were used for acquiring data for building up the Whitelist component. This includes:

- *Automatic collection of named entity instances and indicators from annotated corpora.* The ACE (http://projects.ldc.upenn.edu/ace/) and the ATB (http://www.ircs.upenn.edu/arabic/)[1] are some great resources that facilitate corpus-based studies of many interesting linguistic phenomena in Modern Standard Arabic. These corpora were exploited for the data-collection task. These corpora, which are tagged with many linguistic details, were first analyzed and the commonly occurring patterns studied. These identified patterns were then used to extract useful data.
- *Acquisition of named entities from a database provided by a government organization.* The person and company-name dictionaries also were built from names collected from some organizations including immigration departments, educational bodies, and brokerage companies.
- *Automatic acquisition of named entities from Internet resources.* Names were retrieved further from various Web sites[2] containing lists of Arabic names, company names, and locations. Some of these names are Romanized (written using the Latin alphabet) and had to be transliterated from English to Arabic.

Once NEs were compiled from the corpora processing, Internet resources, and various organizations, they had to be further processed to ensure that the compiled data were clean. The raw data received had to be further processed to make it suitable for incorporation into the system.

[1]Both software systems are available to BUiD under license agreement.

[2]Web sites included: 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

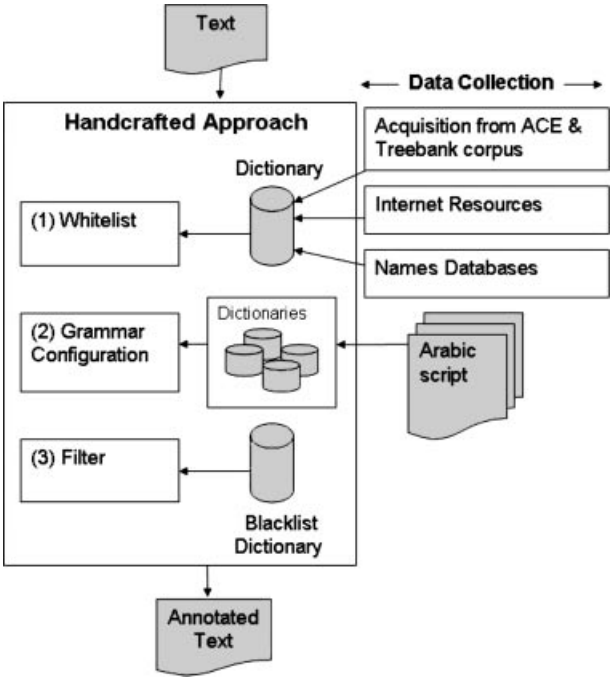

FIG. 1. Architecture of the System.

Architecture of the NERA System

The NERA system requires two main processing resources: a *Whitelist* (gazetteer) and a finite state transduction *grammar*. A *filtration mechanism* also is employed that enables revision capabilities in the system. Figure 1 shows the abstract architecture of the NERA system. The system converts the unstructured input Arabic text into structured form by producing the annotations of the Arabic NE as a result of the recognition task.

The recognition techniques employed include the following two major steps: (a) a lookup procedure, called *Whitelist*, that performs the recognition based on a gazetteer containing lists of known named entities; and (b) a finite state transducer, called *Grammar Configuration*, based on a set of grammar rules derived by analyzing the local lexical context.

Whitelist

The Whitelist plays the role of fixed static dictionaries of various NEs. It is a mechanism that accepts matches that 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 format:

الشيراوى عبدالرحمن قاسم|Abdulrahman Qasim Mohammed Alshirawi

DOI: 10.1002/asi

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 metadata.

Grammar

The grammar performs recognition and extraction of Arabic named entities from the input text based on derived rules. It describes patterns to match NEs, annotations being created as a result. Due to the peculiarities and complexity of the Arabic language, grammar rules are a vital processing resource for the recognition system. For instance, the lack of capitalization for proper nouns can be largely dealt with by using NE indicators to formulate recognition rules. These NE indicators were obtained as a result of the deep contextual analysis of various Arabic scripts that were performed during the data-collection phase. The indicators are referred to as *trigger words* within our system, forming a window around a named entity, which helps in identifying an NE within text, but does not get recognized itself. The following are examples of indicators used within rules.

- Person title: السيدة (Mrs.), السيد (Mrs.)
- Job title: الدكتورة (the doctor), أستاذ العلوم (the sciences professor)
- Company indicator: ذات مسئوليه محدودة (LLC)
- Country postindicators: الاتحادية (the federal), الديمقراطية (the democracy)
- City postindicators: عاصمة المالية (the finance capital)
- Measurement: ملليجرامات (mg), كيلوا مترات (km)
- Price: جنيه مصري (Egyptian pound), درهم إماراتي (Emirate dirham)

Being an agglutinative language, Arabic has highly inflected forms. Hence, the grammar rules build encoded morphological information that describes the inflected structure of the candidate-word forms. This enables stripping off of the prefixes and suffixes from the word stem, thereby ensuring the recognition of the actual NE instance alone. For example, the following pattern describes an optionally defined (masculine or feminine) adjective derived from a location (DAL): (ال)? + country name + (ي | ية), where country name is an entry in the *country name* dictionary that lists all the country name base forms along with their variants. This pattern can, for example, be used to indicate a company صحيفة الأهرام **المصرية** (Al-Ahram Egyptian newspaper), a person الرئيس **المصري** حسني مبارك (The Egyptian President Hosni Mubarak), a price جنيه **مصري** (Egyptian pound), and so on.

For each type of NE, several rules were built, and each one was applied in a particular order to ensure that the most comprehensive recognition result was achieved. Some examples of NERA's grammar rules are provided next.

Example rule for *Person name* recognition

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

This rule recognizes a person name composed of a first name followed by optional last name based on a preceding person-indicator pattern, or the trigger words.

The following name entities would be recognized by the previous rule:

- الملك عبد الله [The King Abdullah]
- الملك الأردني عبد الله [The Jordanian King Abdullah]
- الملك الأردني عبد الله الثاني [The Jordanian King Abdullah II]
- الملكة الأردنية رانيا [The Jordanian Queen Rania]

Apart from contextual cues, the typical Arabic naming elements were used to formulate rules such as *nasab*, *kunya*, and so on (Shaalan & Raza, 2007). Thus, the rules resulted in a good control over critical instances by recognizing complex entities.

Example rule for *Company name* recognition

```
(و | لل | ال | ل | ب)?
(Company_preceding_indicator|company_
preceding_known_part) + ((unknown + (DAL|
company_following_indicator| ((prefix_
busines)?businessType + (DAL)?))
```

This rule recognizes a company name based on prefix words such as 'شركة' (Company) or 'صحيفة' (newspaper) from the dictionary of company preceding indicators, which form a part of the recognized company name. Additionally, business-type indicators such as 'لخدمات الانترنت' (for Internet services) or suffix words/phrases such as 'ذ م م' (LLC) appearing at the end of the company names aid in the recognition process.

The following named entities would be recognized by the previous rule:

- لشركة بي بي لخدمات الانترنت (for BB company for Internet services)
- وصحيفة الأهرام المصرية (and Al-Ahram Egyptian newspaper)
- شركه جوفيند التجاريه ذ م م (Jofind trading company LLC)

Example rule for *Date* recognition

```
(Weekday)? + ws + (DayFigArabic-Indic|
DayFigArabic) + ws + (( و + (DayFigEng|
DayFigArabic))| من)? + ws + (MonthName(
[-/](MonthName))?) + ws + ([-/] (من | في)?
(سنة | العام)? + ws + (yearFigArabic|
yearFigArabic-Indic|relativeDateWord|
yearWord) (year_range)?)
```

In its primitive format, the previous rule matches dates in the format "day month year" with an optional weekday name at the beginning. The day of month in Arabic dates can be represented in either Arabic-Indic or Arabic digits; both variations are matched by this rule.

Moreover, Arabic dates can optionally be comprised of two different month names representing different calendar

DOI: 10.1002/asi

schemes; one of them is usually from the Gregorian calendar. The rule is capable of recognizing this peculiarity in Arabic dates. Further, the year in Arabic dates can be stated in either words or figures (Arabic-Indic or Arabic numerals), or be represented by a relative word such as 'السابق' (previous). All these variations are dealt with well by the aforementioned rule.

The following name entities would be recognized by the previous rule:

- السبت ١٩ من كانون الثاني/يناير ١٩٩٩ (Saturday, 19th of Kanoun, the 2nd of January 1999)
- السبت 6 كانون الثاني من العام 2002 (Saturday, 6th of January of Year 2002)
- مايو الماضي 24 (24th of last May)
- مايو من سنة 1999 وحتى نهاية عام 2001 (May from Year 1999 till the end of Year 2001)
- 18 و 19 يناير (18th and 19th of January)
- 28 (سبتمبر) ايلول [28th of September (Aylol)]

Example rule for *Location* recognition

```
(((مدينة | Administrative division) + ws)?
+ city name +ws + direction)
```

This rule recognizes a city name (existing in the dictionary of city names). The following name entity would be recognized by the this rule:

مدينة اغادير جنوب ... (Agadir City south of . . .)

Filter

A *filtration* mechanism is used that serves two different purposes: revision of the NE extractor results and disambiguation of matches returned by different NE extractors. The *Revision* capability is based on a *Blacklist* (rejecter) dictionary within the grammar configuration to filter matches, returned by rules that appear before or after NE indicators or trigger words but are invalid entities. These invalid entities are derived by analyzing the local lexical context of named entities during grammar rule formulation. This process is illustrated by 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. In this example, however, the sequence of words following the person indicator, [i.e., 'الامين العام' (the Secretary- General)], is not a valid person name; it acts as an appositive. Hence, the role of the Blacklist, another set of rules, comes into play by rejecting the incorrect matches returned by certain grammar rules.

Apart from the *Blacklist* component, certain heuristic *filter rules* are used for postprocessing the system's extraction results to disambiguate extracted named entities. These rules make it possible to disambiguate matches returned by different NE extractors by heuristic prioritization rules. When applying a set of single-slot extraction rules to the input text (i.e., sets of rules that extract particular types of named entities one after the other), one cannot exclude the possibility of identical or overlapping textual matches within the document, among different rules for different named entities. For instance, different sets of rules for extracting instances of both the named entities *Person* and *Location names* may overlap or exactly match in certain text fragments, resulting in ambiguous named entities. Among these named entities, the correct choice must be made. The *filter rule* is an intelligent way of making the correct choice, with respect to the context in which the ambiguous situation arises. The following example indicates an ambiguous situation in Arabic script:

احمد اباد لديه اهتمام بالغ بالفلسفة (**Ahmed Abad** has a keen interest in philosophy)

In this example, the boldface text fragment "احمد اباد" (Ahmed Abad) represents both a person name and a location name. Hence, when NERA is applied here, both the Person and the Location Extractors within NERA will return matches as "احمد اباد" (Ahmed Abad), thereby giving rise to an ambiguous situation. Sometimes, the required behavior is to have exactly one result. In this case, the following filter rule can be used to disambiguate the aforementioned situation:

*If a possible match M1 for a location entity reported by the location extractor **intersects** with a match M2 of a person entity that is also reported by the person extractor, then the match as a location name will be discarded.*

So, in case of an intersection, the match for person names is preferred over location names. Thus, the filter rules defined within the system play a significant role in handling such situations and resolving ambiguity.

FAST ESP—NERA Implementation Platform

The NERA system was implemented and incorporated into the FAST ESP framework (FAST, 2008). FAST ESP is an integrated software environment for development and deployment of 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. FAST ESP supports a set of rule-based tools that we used to deploy our system. It also includes the "hurricane" evaluation tool, which we used to perform our NERA evaluation using a reference corpus.

NERA is implemented within the entity-extraction component of the Content Pipeline, in the Document Processing Engine of FAST ESP. Figure 2 indicates the functionality of

DOI: 10.1002/asi

FIG. 2. NERA incorporated into the FAST ESP (2008) pipeline to recognize named entities in text.

the NERA system incorporated in the pipeline within FAST ESP for recognizing and tagging named entities in text.

Resources Built for Arabic NER Within NERA

To develop the Arabic NER, we had to build our own corpora due to the unavailability of free Arabic corpora for research purposes. Moreover, the commercially available Arabic corpora are oriented towards the newswire domain, which we found lacks equal coverage of the 10 named entities involved in our research. Further, we also have built the Whitelist (gazetteer) component, which is a vital processing resource for many NLP tasks. In this section, we present the main characteristics of the resources developed for Arabic.

Corpora for Person, Location, Date, Time, Price, and Measurement NEs

ACE (Version 5.3.3 2005.05.31) and ATB (Version 2.0, LDC Catalog No. LDC2003T06) corpora are standard Arabic resources built by LDC for Arabic NLP tasks. These corpora mainly contain text taken from newswire documents and broadcast news which was used to create the entity tagged reference corpora for evaluating the following extractors: Person, Location, Date, Time, Price, and Measurement within NERA.

The tagset used by LDC within these corpora provides very detailed and sophisticated annotation, with markup based on Arabic linguistics associated with the Arabic language. Using Python scripts and a pattern-matching algorithm, we first acquired NEs from LDC's original tagset. For instance, for extracting Person NEs, the script was programmed to match the "PER" tag within the ACE corpus and the "PropNoun" tag within the ATB corpus. The acquired NEs were then used to create our NE tagged reference corpora, with 10 different tagsets (e.g., `<person>...</person>` tags for Person NE). The tagging was done in a semi-automated way as follows:

- The *Person names* and *Locations* contained within the Source Arabic Text from ACE and ATB was automatically tagged using Python scripts and the acquired NEs.
- The same reference corpus was further *hand-tagged* again to mark the *Date*, *Time*, *Price*, and *Measurement* NEs. The manual tagging was done for two reasons:
- The tagset in ACE and ATB uses a generic POS tag "Numeric" for entities such as price, measurement, and percentages.
- A common tag "TIMEX" is used by ACE and ATB to tag both *date* and *time* entities in a combined way.

For efficiency, the reference corpus that we built was divided into sets of test corpora, each being approximately 100 KB in size. The total number of test sets for these named entities is 34, with 24 created from the ACE corpus and 10 created from the ATB corpus. The total size of the reference corpus is around 4 MB, composed of 300,000 distinct words. The size and content of the corpus are such that it contains a representative number of occurrences of the following NE types: The *person name* category includes 500+ entities, the *location* category includes 500+ entities, the *date* category includes 394 entities, the time category includes 110 entities, the price category includes 400 entities, and the measurement category includes 386 entities.

Corpus for Company-Named Entities

The ACE and ATB corpora do not include a representative number of entities for company names. Thus, we sought another corpus, the Corpus of Contemporary Arabic (CCA[3]), to create the reference corpus for evaluating the *company extractor*. This choice was based on the fact that the text within CCA gave a good, varied coverage of company names, thereby ensuring a more reliable evaluation of the company extractor. For building up the company test corpus, we created two reference corpus sets (each 100 KB in size) from randomly selected text from the CCA corpus. Both sets were hand tagged to mark company names within them. A total of 226 company-name instances have been tagged.

Named Entity Corpus for Phone Numbers, ISBNs, and File Names

Available corpus resources in Arabic are quite limited and restricted to coverage of the most important NEs such as Person, Location, and so on. Hence, various Arabic Web sites (e.g., Real Estate, Newspaper, etc.) were analyzed to collect text containing phone number, ISBN, and file-name entities. The corpus built was hand-tagged with 191 phone number entities, 100 entities for ISBNs, and 139 entities for file names.

[3]CCA can be freely downloaded online from Latifa Al-Sulaiti's web site, http://www.comp.leeds.ac.uk/eric/latifa/research.htm. As indicated by the developers, the Arabic text within this corpus was mainly acquired from magazine and newspaper web sites.

DOI: 10.1002/asi

In summary, the reference corpora for evaluating the 10 types of named entities (person, location, company, date, time, price, measurement, phone number, file name, and ISBN) within NERA are divided in the following way:

- 34 corpus sets for person, location, date, time, price, and measurement extractor evaluation (created from ACE and ATB corpus text)
- 2 sets of corpora for company extractor evaluation (created from CCA corpus text)
- 3 individual reference corpora each for phone number, ISBN, and file name extractor evaluation (created from text at various Arabic Web sites)

The corpora created are in the XML format with UTF-8 encoding, in accordance with the guidelines set forth at the beginning of the project. Additionally, the size and content of the corpora are such that they contain a representative number of occurrences of all 10 entity types.

Whitelist/Dictionaries Built

NERA gathers three different manually built gazetteers:

- *Person gazetteer*: This contains a list of 263,598 complete names of people collected from various government organizations, existing Arabic corpora, and Internet resources. Further, the names were split into dictionaries of first and last names, omitting the repeated names; the final list contains 175,502 first names and 33,517 last names.
- *Location gazetteer*: This consists of 4,900 names of continents, countries, cities, states, political regions, towns, and villages found in the Arabic version of Wikipedia and other Web sites.
- *Organization gazetteer*: This consists of a list of 273,491 names of companies, including those in areas of media and newspapers, construction, banks and insurance, airlines, and telecommunications, among others.

Experiment

The evaluation of the NERA extractors was performed using our own reference corpora, which highlight the Arabic resources built during this project.

As mentioned in the previous section, the *Whitelist* built for Person, Location, and Company NE extractors contains certain entries extracted from the ACE and ATB corpora. The evaluation corpus, to some extent, was built using the same Arabic corpora resources; however, since the corpora were huge in size, the overlap between texts used for Whitelist and evaluation corpora building was kept minimal. Additionally, the positive recognition results achieved can be attributed mainly to the grammar rules, as compared to the gazetteer, since the pattern matching developed was able to deal with issues peculiar to the Arabic language, including inflections, typographic variation, and so on.

The Evaluation Method

The performance was measured by *Precision, Recall, and F-measures,* which are the standard measures for NER (De Sitter, Calders, & Daelemans, 2004):

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

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

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Another way to look at Precision and Recall is:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

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, an average accuracy is often reported in the form of the *F-measure*, a harmonic mean that equally weights recall and precision. It was introduced to provide a single figure to compare different systems' performances.

Since the corpora were tagged in a semi-automated way, certain named entities were left untagged. In the recognition results, these NEs were recognized correctly by the system, but since they were not tagged in the test corpora, the evaluation tool marked these entities as *false positives* when in reality they were *true positives*. To overcome this issue, the entities marked as false positives by this tool were identified and retagged (i.e., manually corrected) in the reference corpora. This iterative tagging of the corpus ensured quality.

The NERA system implemented within the FAST ESP pipeline was evaluated using an information-extraction-testing tool called *Hurricane* that applies the aforementioned standard measures. This tool can perform evaluation on a corpus with a size limit to 100 KB. Hence, the 5 MB of evaluation corpora built were divided into 46 sets of corpus files. Each test set was then individually given as input to *Hurricane*, and separate accuracy results were produced by each. The average of the results was estimated to reach conclusions about each NE's recognition accuracy.

Results

At the beginning of the project, we set minimum acceptance criteria based on previous experience of FAST gained from various NER systems for languages other than Arabic. Table 5 shows the comparison between the achieved accuracy and these minimum acceptance criteria (Excellent = 90–100%, Good = 80–89%, Fair = 70–79%, Poor = <70%). From this table, note that the precision achieved is almost the same as planned whereas the recall achieved was higher than that of the initial plan.

DOI: 10.1002/asi

TABLE 5. Comparison of accuracy achieved and acceptance criteria set.

	Acceptance criteria		Achieved accuracy	
Entity type	Precision	Recall	Precision	Recall
Person name	Good	Fair	Good	Good
Organization	Fair	Fair	Good	Good
Locations	Fair	Fair	Fair	Good
Date	Excellent	Good	Excellent	Excellent
Time	Excellent	Good	Excellent	Excellent
Price	Excellent	Good	Excellent	Excellent
Measurements	Excellent	Good	Excellent	Excellent
Phone no.	Excellent	Good	Excellent	Good
ISBN	Excellent	Good	Excellent	Excellent
File name	Excellent	Good	Excellent	Excellent

TABLE 6. Accumulated accuracy of the 10 NEs.

No.	Entity type	Precision (%)	Recall (%)	F-measure (%)
1	Person	86.3	89.2	87.7
2	Location	77.4	96.8	85.9
3	Company	81.45	84.95	83.15
4	Date	91.2	92.3	91.6
5	Time	97.25	94.5	95.4
6	Price	100	99.45	98.6
7	Measurement	97.8	97.3	97.2
8	Phone no.	94.9	87.9	91.3
9	ISBN	94.8	95.8	95.3
10	File name	95.7	97.1	96.4

Table 6 summarizes the accumulative recognition accuracy, in terms of precision and recall, achieved by each of the 10 extractors built within NERA against the reference corpora.

With respect to the results of the extractors handling the person, location, and company types, some of the entries within the Whitelist component built were extracted from the same corpus also used for creating the reference corpora for evaluation. However, the evaluation results achieved are accurate since they indicated recognition of named entities not included in the Whitelist but being recognized by the grammar rules within the pattern-matching component. After careful analysis of the evaluation results, we found that the accuracy can be further improved in the following ways:

- Expanding the Whitelist dictionary of Person, Location, and Company Names further.
- More Arabic text/corpora can be analyzed to identify strings that act as named entity indicators.
- Reducing negative effects on evaluation results (e.g., true positive being treated as false positives) because of incomplete annotation of the test corpora. The reference corpora can be further fine-tuned to tag each and every named entity instance.
- Enhancing the quality of transliterated names used.
- Using Arabic text with error-free spelling.
- Including all possible spelling variations used for names in Arabic written text in an automated way.

One important factor that has greatly influenced the results achieved is the nonstandardization of written Arabic text. The majority are unstructured and are loaded with inconsistencies due to the lack of control over written forms of Arabic script. Standard practices in publishing written Arabic resources can help achieve far better accuracy results.

Conclusion

Arabic is a relatively complex and difficult language to analyze, not so much because of its difficult morphological structure but mostly because of how that structure is impacted and made more complex by the orthographic issues of its written form coupled with the drawbacks of limited research done for the Arabic language. This work is an attempt to broaden the coverage for entity extraction incorporating the Arabic language by overcoming the language-specific challenges to a great extent, thereby paving the path towards enabling search solutions for the Arabic market.

Various data-collection techniques were used for acquiring dictionary name lists. The rule-based approach employed with great linguistic expertise led to a successful implementation of the NERA system by overcoming the challenges posed by Arabic language. A set of grammar rules was derived by analyzing the local lexical context of a large amount of diverse data. 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 nonname context begins. Further, the

DOI: 10.1002/asi

intelligent use of filter rules helps in dealing with recognition ambiguity between named entities. We have evaluated our system performance using our own corpora tagged in a semi-automated way. Moreover, these corpora could be used as a standard evaluation dataset for Arabic NER approaches. The evaluation results thus far look very promising; NERA achieved high average precision and recall for each named entity type against the reference corpora. Suggestions for improving the system performance based on analyzing the results were provided.

Acknowledgment

This work was funded by the "Named Entity Recognition for Arabic" joint project between The British University in Duabi, Dubai, United Arab Emirates and FAST Search & Transfer Inc., Oslo, Norway. FAST was recently acquired by Microsoft. We thank the FAST team; in particular, 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

Abuleil, S. (2004). Extracting names from Arabic text for question-answering systems. In Proceedings of the 7th International Conference on Coupling Approaches, Coupling Media, and Coupling Languages for Information Retrieval (pp. 638–647), University of Avignon (Vaucluse), France.

Benajiba, Y., Diab, M., & Rosso, P. (2008). Arabic named entity recognition: An SVM-based approach. In Proceedings of 2008 Arab International Conference on Information Technology (ACIT) (pp. 16–18). Amman, Jordan: Association of Arab Universities.

Benajiba, Y., & Rosso, P. (2008). Arabic named entity recognition using conditional random fields. Proceedings of the Workshop on HLT & NLP Within the Arabic World. Arabic Language and Local Languages Processing: Status Updates and Prospects, 6th International Conference on Language Resources and Evaluation (pp. 26–31). Marrakech, Morocco.

Chinchor, N. (1998). Overview of MUC-7. In Proceedings of the 7th Message Understanding Conference (pp. 2–5). Retrieved January 26, 2009, from http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Crestan, E., & de Loupy, C. (2004). Browsing help for a faster retrieval. Proceedings of the 20th International Conference on Computational Linguistics (pp. 576–582). New York: ACM Press.

De Sitter, A., Calders, T., & Daelemans, W. (2004). A formal framework for evaluation of information extraction. University of Antwerp, Department of Mathematics and Computer Science, Technical Report TR 2004–0. Retrieved April 15, 2009, from http://www.cnts.ua.ac.be/Publications/2004/DCD04

FAST. (2008). FAST ESP: The world's most intelligent, secure, high-performance search platform. FAST, A Microsoft Subsidiary in Oslo, Norway. Retrieved January 26, 2009, from http://www.fastsearch.com/l3a.aspx?m=1031

Gey, F. (2000). Research to improve cross-language retrieval. Position paper for CLEF. In C. Peters (Ed.), Proceedings of the cross-language information retrieval and evaluation. Workshop of Cross-Language Evaluation Forum, Lisbon, Portugal. Lecture Notes in Computer Science, 2069 (pp. 83–88). Berlin: Springer.

Larkey, L., Abdul Jaleel, N., & Connell, M. (2003). What's in a name? Proper names in Arabic cross language information retrieval. CIIR Technical Report No. IR-278. Retrieved January 26, 2009, from http://ciir.cs.umass.edu/pubfiles/ir-278.pdf

Maloney, J., & 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 (pp. 8–15). Montreal.

Samy, D., Moreno, A., & Guirao, J. (2005). A proposal for an Arabic named entity tagger leveraging a parallel corpus. Proceedings of the International Conference on Recent Advances in Natural Language Processing (pp. 459–465). Borovets, Bulgaria: Benjamins.

Shaalan, K. (2005). Arabic GramCheck: A grammar checker for Arabic. Software Practice and Experience, 35(7), 643–665. Chichester, England: Wiley.

Shaalan, K., & Raza, H. (2007). Person name entity recognition for Arabic. Proceedings of the ACL 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources (pp. 17–24). Prague, Czech Republic: Association for Computational Linguistics.

Shaalan, K., & Raza, H. (2008). Arabic named entity recognition from diverse text types. In B. Nordström & A. Ranta (Eds.), Proceedings of the 6th International Conference on Natural Language Processing, Gothenburg, Sweden. Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI): Advances in Natural Language Proceedings (Vol. 5221, pp. 440–451). Berlin, Germany: Springer-Verlag.

Zitouni, I., Sorensen, J., Luo, X., & Florian, R. (2005). The impact of morphological stemming on Arabic mention detection and coreference resolution. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, 43rd annual meeting of the Association of Computational Linguistics (pp. 63–70). Ann Arbor, MI: Association for Computational Linguistics.

DOI: 10.1002/asi

Appendix

Dictionaries of NE Indicators

The following dictionaries were derived using the aforementioned data-collection techniques. The following specifies various indicator dictionaries along with their respective number of entries for eight types of named entities covered by NERA. The other two types of named entities that do not have indicator dictionaries and rely only on grammar rules are ISBN and file name.

Table A1. Dictionaries of NE indicators.

Person	Company	Location	Date	Time	Price	Measurement	Phone no.
Job titles (19,245)	Business type (1,410)	Administrative division (23)	Month names (156)	Time zones (13)	Currency name (37)	Unit 1 (481)	Phone indicators (160)
Person titles (20)	Company following known part (114)	City preindicators (12)	Weekday (23)	Time word (37)	Power of 10 in words (13)	Unit 2 (14)	Phone-related words (32)
Honorifics (173)	Company following indicator (37)	City postindicators (10)	Related words (11)	Time units (39)	Locations (39)	Rejecter units (1,579)	
Country names (923)	Company preceding known part (163)	Country pre-indicators (77)	Days in word 1–31 (118)	Tens in word (20)			
Laqabs[a] (8,169)	Company preceding indicator (4)	Country postindicators (22)	Hundreds in words (18)	fractions (13)			
Person indicators (421)	Location names (4,909)	Location Blacklist (167)	Tens in words (20)				
	Business prefix (4)	Direction1 (17)	Units in words (43)				
	Company rejecters (4,980)	Direction2 (8)	Date rejecter words (546)				
	Company part rejecters (4,997)	Direction3 (4)					
	Normalization base form (22)	Direction4 (4)					
		Direction5 (4)					
		Location base form (534)					
		Location inhabitant (44)					

[a]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). *Laqabs* follow the *ism*: *Harun al-Rashid* (Aaron the Rightly-guided)].

DOI: 10.1002/asi