What in the World is a *Shahab*? Wide Coverage Named Entity Recognition for Arabic

Luke Nezda, Andrew Hickl, John Lehmann, and Sarmad Fayyaz

Language Computer Corporation
1701 N. Collins Blvd.
Richardson, TX 75080, USA
{luke,andy,john,sarmad}@languagecomputer.com

Abstract

This paper describes the development of CICEROARABIC, the first wide coverage named entity recognition (NER) system for Modern Standard Arabic. Capable of classifying 18 different named entity classes with over 85% F, CICEROARABIC utilizes a new 800,000-word annotated Arabic newswire corpus in order to achieve high performance without the need for hand-crafted rules or morphological information. In addition to describing results from our system, we show that accurate named entity annotation for a large number of semantic classes is feasible, even for very large corpora, and we discuss new techniques designed to boost agreement and consistency among annotators over a long-term annotation effort.

1. Introduction

Named entity recognition (NER) systems can be used to provide a wide range of information about the different types of entities mentioned in texts. By classifying entities in texts with regards to an ontology of semantic types, NER systems unlock a wealth of semantic information that can be used in sophisticated natural language processing applications, such as automatic question answering (Harabagiu et al., 2001), information extraction (Surdeanu and Harabagiu, 2002), and multi-document summarization. As system developers seek to extend current NLP technologies to languages of interest such as Arabic, the need for reliable and accurate NER systems is greater than ever. For example, an NER system for Arabic could be used to obtain information about the people, organizations, locations, and quantities mentioned in the text in Table 1, without the need for an automatic machine translation system.

English	The suspects in the cases have no links to **[al Qaeda]**$_{\text{organization}}$, led by **[Usama bin Laden]**$_{\text{person}}$, or **[Jihad Group]**$_{\text{organization}}$, the Egyptian organization headed by **[Ayman al-Zawahiri]**$_{\text{person}}$, and it has been proven that the **[two suspects]**$_{\text{count}}$ trained to be pilots in the **[United States]**$_{\text{location}}$, but had nothing to do with the attacks on **[New York]**$_{\text{location}}$ and **[Washington]**$_{\text{location}}$.
Arabic	تأكد أن المتهمين في القضيتين لا علاقة لهم بتنظيم [القاعدة]$_{\text{organization}}$ الذي يقوده [أسامة بن لادن]$_{\text{person}}$ أو جماعة [الجهاد]$_{\text{organization}}$ المصرية التي يتزعمها الدكتور [أيمن الظواهري]$_{\text{person}}$ إذ تبين [أن اثنين]$_{\text{count}}$ من المتهمين في القضية تلقيا تدريبات على الطيران في [أمريكا]$_{\text{location}}$ وليست لهما أي علاقات بالهجمات في [نيويورك]$_{\text{location}}$ و [واشنطن]$_{\text{location}}$.

Table 1: Named Entity Recognition in Arabic and English

Despite this great potential, most NER systems have focused on classifying only a select few types of entities: systems developed for the 1996 and 1997 Message Understanding Conferences (MUC-6 and MUC-7) focused on a maximum of 8 entity types, while systems built as part of the 2002 and 2003 Conference on Natural Language Learning (CoNLL) Shared Task classified only a set of 4 types. Although some of the systems developed for these evaluations have achieved relatively high levels of performance in classifying person, location, and organization names, their limited coverage has prevented them from providing the full range of semantic annotations NLP applications need to be able to find and extract information automatically from texts.

The coverage of NER systems has traditionally depended on access to large-scale lexical resources such as named entity grammars, lexicons, and sources of annotated training data. While heuristic-based NER systems (Appelt et al., 1995) were able to achieve nearly 90% F for certain entity types, the performance of these systems was often limited by the size and quality of the rules created for each individual entity type. Increasing the coverage of these systems proved challenging, as developers had to create new type- and language-specific grammars for each additional class of entities that was to be recognized. In contrast, while supervised machine learning-based approaches to NER (Bikel et al., 1997; Cucerzan and Yarowsky, 1999; Klein et al., 2003) have enabled the development of accurate NER systems without the need for specialized heuristics, the expansion and refinement of these systems has been shown to be dependent on their access to sources of high-quality annotated training data. In order to build wide coverage NER systems using these approaches, system builders must be able to annotate very large corpora reliably and consistently with a significant number of semantic types. Finally, even unsupervised approaches to NER (Pasca, 2004) face significant resource limitations. Although these approaches avoid the need for precoded grammars or sources of annotated training data, they do depend on access to large amounts of data and sets of "seed" patterns that are used to find entities in text.

Despite these limitations, we feel that developers of NER applications should focus on building systems that are capable of classifying as wide of a range of semantic types as possible. In order to demonstrate that the creation of an accurate, wide coverage NER system is feasible utilizing currently available techniques, we have developed a new NER system for Arabic, known as CICEROARABIC, which is capable of recognizing a total of 18 different named entity classes with over 85% F. We describe how we met three

different challenges to building a wide coverage NER system for Arabic. First, we present a new wide coverage named entity annotation schema that can be used for any human language. Second, we discuss the quality control techniques that we used in order to maximize annotation quality over a 4 month annotation effort that resulted in the tagging over 800,000 words of Arabic newswire text. Finally, we show how we used CICEROARABIC's very large annotated corpus to achieve high wide coverage NER performance for without the need for any morphological features or specialized lexica.

The rest of this paper is organized in the following way. Section 2 discusses previous approaches to the problem of NER. Section 3 presents a new 18-class named entity annotation for Arabic and discusses the methodology we employed in annotating an 800,000-word corpus with named entity information. In Section 4, we describe the architecture of the CICEROARABIC Arabic NER system. Section 5 describes the performance of this system, and Section 6 presents our conclusions.

2. Previous Work

Work in NER has traditionally focused on three basic types of approaches: (1) heuristic-based systems, which recognize and classify named entities in text using sets of hand crafted rules and patterns, (2) supervised machine learning-based approaches which use large sources of annotated training data to train classifiers which can identify different types of named entities, and (3) unsupervised approaches, which derive named entity classifications from large corpora without the need for additional hand-coded rules and/or additional annotations.

Early approaches to NER (Appelt et al., 1995) used sets of hand-crafted grammar rules in conjunction with cascades of finite-state automata in order accurately identify and classify named entities in text. While these kinds of heuristic-driven approaches achieved levels of performance near to that of human annotators, the overall performance of these systems depended largely on the creation of knowledge-intensive resources that required months of skilled labor to develop.

In contrast, work done by (Bikel et al., 1997; Cucerzan and Yarowsky, 1999; Klein et al., 2003) has shown that supervised machine learning-based techniques can be used to accurately recognize named entities in texts without the need for type-specific rules or lexica. While a number of machine learning algorithms have been employed (including decision trees, Maximum Entropy, and Support Vector Machines), these approaches have traditionally cast the problem of named entity recognition as a classification problem which depends on access to large sources of training data.

Recent work (Thelen and Riloff, 2002; Pasca, 2004) has proposed that a family of unsupervised approaches to NER can be used to counter the "knowledge bottleneck" faced by previous heuristic-based or supervised machine learning-based approaches. Work done by (Thelen and Riloff, 2002) demonstrated that co-training techniques, coupled with a small set of seed examples, could be used to automatically learn how types of NE should be classified. In a similar fashion, (Pasca, 2004) utilized a relatively small number of high-precision extraction patterns in order to gather thousands of open-domain and non-disjoint lexicons from large web corpora.

While unsupervised methods appear promising in terms of their overall resource requirements and their portability to new domains and/or languages, we believe that supervised machine learning methods currently still represent the best approach for creating accurate named entity recognition systems. In the next section, we introduce a novel set of named entity classification guidelines and describe how we used them to create a large annotated corpus for named entity recognition in Arabic.

3. Named Entity Recognition in Arabic

Arabic has been the focus of much recent work in the field of natural language processing. Facilitated by the release of large Arabic corpora, including the Penn Arabic Treebank (Maamouri et al., 2004) and the Arabic Gigaword corpus, researchers have now developed a number of high-performance and readily available text processing tools, including tokenizers, part-of-speech taggers, phrase chunkers, morphological analyzers, and syntactic parsers. These tools are now setting the stage for the development of a new generation of sophisticated NLP tools for Arabic, including automatic question-answering, information extraction, and named entity recognition systems.

Even though an Arabic entity detection and recognition (EDR) task was included in both the 2004 and 2005 Automatic Content Extraction (ACE) evaluations, work in Arabic NER remains in its infancy. In early work, (Maloney and Niv, 1998) described the development of an Arabic NER system, known as TAGARAB, which utilized morphological features in conjunction with name-finding heuristics to recognize 5 named entity types with 85% F on a small newswire corpus. To our knowledge, no other published work has tackled the challenge of performing NER in Arabic.

In the past, part of the challenge of performing accurate NER for Arabic was the lack of a large corpus annotated with named entity information. As part of the 2005 ACE evaluation, the Linguistic Data Consortium released a 133,000 word mixed-genre corpus that was annotated with 7 entity types and a total of 45 different entity subtypes. While the creation of this resource represents an important step forward, we feel that still larger corpora are needed to develop the types of robust, domain-independent NER systems that form the core of sophisticated NLP applications. In this section, we describe the creation of a new 800,000-word annotated corpus for the development of Arabic NER systems. This corpus, which we refer to as the CICEROARABIC NER Corpus includes annotations for 18 different named entity types derived from the past named entity annotation guidelines prepared for past NER evaluations such as the Message Understanding Conferences (MUC), the DARPA TIDES (Translingual Information Detection, Extraction, and Summarization) program, and the annual ACE evaluations. In Section 3.1, we describe our new annotation schema, while in section 3.2, we describe how our annotation efforts in detail.

3.1. Guidelines for Entity Annotation

We believe that there are 3 criteria that any successful named entity annotation schema must satisfy. First, the NE classes selected for a schema should represent a set of natural kinds that can be reliably distinguished by human annotators. Care should be taken to avoid introducing classes where the assignment of an NE type depends (even partially) on an annotator's interpretation of a context or personal knowledge of a specific domain. Second, NE classes should represent a set of separable types. Since most NER applications assign only a single class to individual entity, annotation guidelines should not include types that contain substantial overlap in their membership. Finally, NE schema should ultimately be manageable: while increasing the number of NE types that an NER system can classify does have value for the development of open-domain text processing systems, we recognize that individual annotators can accurately annotate only a limited number of types simultaneously. Increasing the annotation load beyond a certain point can only result in degradation of overall annotation quality, both in terms of precision and recall.

Over the past 10 years, a number of NE guidelines have been introduced to evaluate NER systems. Among the first were the MUC-6 and MUC-7 guidelines for English (Chinchor, 1997), which distinguished between as many as 8 different NE types, including *person*, *organization*, *location*, *date*, *absolute time*, *relative time*, *money*, and *percent*. These were updated in the ACE Entity Mention Detection Guidelines (Lin, 2005), which focused on a total of 7 named entity types that included *person*, *organization*, *geopolitical entity* (GPE), *(geographic) location*, *facility*, *vehicle*, and *weapon*. Most recently, the Conference on Natural Language Learning (CoNLL) 2002 and 2003 shared task focusing on multilingual NER reduced the total number of types to 4: *person*, *organization*, *location*, and a fourth type, *miscellaneous*, which represented a broad class of proper nouns that included both entities and events.

In this work, we selected a total 18 different named entities that should be recognized and classified in Arabic newswire texts. These 18 classes are organized into 5 subcategories: numeric expressions, temporal expressions, quantities, names, and artifacts.

Numeric Expressions. Six types of numeric tags were defined: number, percent, temperature, money, age, and unit.

Number: Used with numbers that do not necessarily denote a count of objects, such as phone numbers, passport numbers, sports scores, and cardinal designations.
Percent: Used only with percentages. Equivalent to MUC-6 tag ⟨percent⟩.
Temperature: Used only with absolute temperatures. May include "degrees" and modifiers such as "Fahrenheit", "Celsius", "Kelvin", "above zero", or "below zero". Relative temperatures (e.g. "25 degrees higher") were not tagged.
Money: Used only with absolute monetary amounts. Equivalent to MUC-6 tag ⟨money⟩.
Age: Used only with explicit mentions of the absolute age of entities. includes value and unit (when available).
Unit: Used with all standard and metric units. Includes value and unit (when available).

Figure 1: Numeric Expressions

Temporal Expressions. Three types of temporal expressions were tagged: exact dates, exact times, and time quantities. While *exact date* and *exact time* were equivalent to the MUC-6 ⟨date⟩ and ⟨time⟩ tags, *time quantity* was added to capture expressions referring to an inherent duration of time.

Exact Date: Used with mentions of specific dates. Must refer to a particular absolute calendar day, date, month, season, century, or period. Equivalent to MUC-6 tag ⟨date⟩.
Exact Time: Used with a mention of a specific clock time.
Time Quantity: Used to tag expressions referring to specific delimited periods of time that can be identified using specific reference points.

Figure 2: Temporal Expressions

Quantities. Two types of quantity tags were defined: *person count*, used with quantities of humans, and *other count*, used with quantities of all other countable items.

Person Count: Used only with numbers denoting counts of humans. Can be used if label is not present, or is inferable from coreference.
Other Count: Used with numbers denoting counts of any non-human items.

Figure 4: Quantity Types

Names. Four types of proper name tags were defined: person, organization, political location, and geographic location. All location names should be tagged as (political) locations, even if they're being used to refer to national teams or other organizations that represent the entire country.

Person Name: Used with names of people. First names, middle names, nicknames, and last names can be tagged together, if adjacent. Equivalent to MUC-6 tag ⟨person⟩.
Organization Name: Used with names of organizations, including government, military, educational, non-profit, and arts organizations. Equivalent to MUC-6 tag ⟨organization⟩.
Political Location: Used to tag names of countries, cities, provinces, states, etc. Not equivalent to MUC-6 tag ⟨location⟩.
Geographic Location: Used with other non-political location names. Used with names of rivers, mountains, natural landmarks, valleys, oceans, seas, lakes, forests, stars, galaxies, etc. Not equivalent to MUC-6 tag ⟨location⟩.

Figure 5: Types of Names

Artifacts. Three kinds of artifact tags were defined: structures, vehicles, and weapons.

3.2. Annotating the Arabic Treebank

We used the named entity annotation guidelines outlined in the previous section to annotate an 806,065 word Arabic corpus consisting of the Penn Arabic Treebank and the Prague Arabic Dependency Treebank (Hajic et al., 2004). Although our approach to NE annotation was not novel, our methodology enabled us to perform a large amount of annotation work quickly and with relatively high inter-annotator agreement. Annotation of this corpus was conducted over a 16 week period by a team of 6 native speakers of Arabic who were also bilingual speakers of English. [1] Each document in the corpus was subject to three separate annotation passes. First, individual documents were annotated by two

[1] Annotators came from 5 different Arabic-speaking countries: Iraq, Jordan, Libya (2), Lebanon, and Syria.

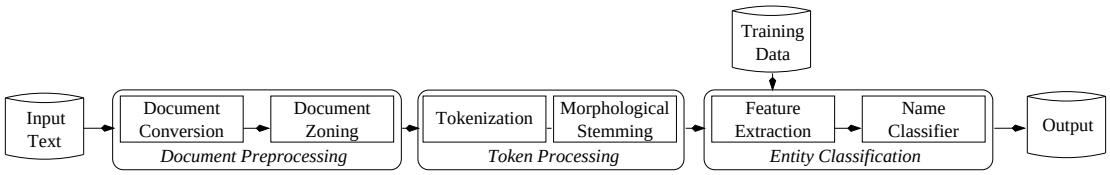

Figure 3: Architecture of the CICEROARABIC Named Entity Recognition System

Facility: Used to tag human-created structures and/or facilities (e.g. buildings, aiports, stadia, houses of worship, etc.).
Vehicle: Used to tag proper names of vehicles.
Weapon: Used to tag common names and trade names of guns, missiles, bombs, chemical weapons, biological weapons or explosives.

Figure 6: Artifacts

annotators each; once annotation was complete, pairs of annotators would confer to compare annotations and resolve any discrepancies. Documents were then passed to a third annotator who manually checked each document for accuracy and completeness. In order to evaluate inter-annotator agreement, a total of 10,000 documents were held out and annotated by 2 different pairs of annotators. We found that post-annotation conferencing significantly improved agreement. Prior to conferencing, average inter-annotator agreement (between individuals) was 87%; this number jumped to 94% (between groups) following conferencing. In the following section, we describe how we used this large corpus of named entity annotations to train a machine learning-based named entity recognition system for Arabic.

4. System Description

This section describes the architecture of the CICEROARABIC named entity recognition system. As can be seen in Figure 3, CICEROARABIC consists of three modules: (1) a *document preprocessing* module which prepares documents for later processing, (2) a *token processing* module which tokenizes Arabic texts and stems individual tokens, and (3) an *entity classification* module which utilizes a Maximum Entropy-based classifier to tag named entities with one of 18 different named entity classes.

CICEROARABIC begins the process of NER for Arabic by converting documents written in Arabic to the Unicode UTF-8 character encoding. During this process, documents written using earlier standards (such as the ISO88959-1, ISO88959-2, ISO8895-6, or the MS 1256 standards) are also detected and converted to Unicode. Documents are then sent to a *document zoning* module, where whitespace and structural cues are used to segment the text into paragraphs and to separate headlines and datelines from the body text of the document.

Once preprocessing is complete, individual documents are then sent to a *tokenization* module, which used a set of heuristics to identify the boundaries of each individual token. In Arabic, orthographic words may include a set of morphologically-bound lexical items (known as *clitics*) which must be segmented during tokenization. Since Arabic words can include as many as two prefixal proclitics and up to one suffixal enclitic, we used tokenization heuristics based on whitespace and punctuation in conjunction with heuristics first introduced in (Larkey et al., 2002)'s *light8* morphological stemmer in order to best approximate the ideal tokenization for a text. After sentences were segmented into sets of words based on whitespace and punctuation, words were normalized by removing diacritics and other non-letter, non-numeral characters. Following (Larkey et al., 2002), clitic prefixes and suffixes (including the definite determiner ال- '*the*' and the conjunction و- '*and*') were removed if their removal left a token that was at least two characters long. Although we have expeimented with the Support Vector Machine-based tokenizers and part-of-speech taggers developed by (Diab et al., 2004), we have found that features derived from our knowledge-lean approach to tokenization and stemming actually improves the overall performance of our NER system.

Tokens were then sent to a *feature extraction* module that was used to compute features for entity recognition from the more than 800,000 words of annotated training data assembled for Arabic. Six classes of features were used in CICEROARABIC: (1) *word-based* features equal to the entire unstemmed lemma of the current word (or the word preceding or following a term), (2) *stemmed word-based* features equal to the *light8* stemmed form of the current word (or the word preceding or following a term), (3) *bigram-based* features equal to the lemma of the pair of unstemmed words preceding and following a term, (4) *prefix-based* features equal to the first n characters of the term being classified, (5) *suffix-based* features equal to the last n characters of the term, and (6) *previous class-based* features equal to named entity class assigned to the previous n tokens. Figure 7 lists the complete set of features used in CICEROARABIC.

Features were then sent as input to a Maximum Entropy-based *entity classifier* which assigned each candidate token a label corresponding to (1) its position within a named entity and (2) its named entity classification. CICEROARABIC uses the standard IOB-style notation to indicate the boundaries of an entity expression: *B* labels are assigned to tokens that mark the beginning of an entity expression, while *I* labels mark tokens internal to an entity expression, and *O* labels denote tokens not deemed to be part of any entity expression. Entities that are assigned either an *I* or *B* label are also a named entity class corresponding to one of the 18 named entity categories described in Section 3.1.. For example, the context containing the three token entity expression اسَمة بن لَدن '*Usama bin Laden*' found in Table 1 was annotated as presented in Table 2. Since Arabic is read right-to-left, the system would begin by first assigning the rightmost token أَسَمة '*Usama*' the label *B-person*, corresponding to the beginning of an entity expression of type PERSON. The system would then consider the next token بن '*bin*' and assign it either an *I-person* label (if it considered the token to be a continuation of the named entity), an *O* label (if it considered the token to not be a named entity),

1. **Word-based Features**: Equal to unstemmed lemma of word; Computed for Word(n-2), Word(n-1), Word(n), Word(n+1), Word(n+2)

2. **Stemmed Word-based Features**: Equal to *light8* stemmed lemma of word; Computed for Word(n-2), Word(n-1), Word(n), Word(n+1), Word(n+2)

3. **Bigram-based Features**: Equal to stemmed bigram lemmas; Computed for bigrams consisting of (Word(n-3) Word(n-2)), (Word(n-2) Word(n-1)), (Word(n-1),Word(n)), (Word(n),Word(n+1)), (Word(n+1) Word(n+2)), (Word(n+2) Word(n+3));

4a. **Prefix-based Features**: Equal to first string of characters ($1 \leq$ n) in unstemmed word; Computed for n $\leq$ 6;

4b. **Prefix-based Features (skip character)**: Equal to first string of characters ($1 \leq$ n) in unstemmed word, skipping the mth character. Computed for n $\leq$ 6, m $\leq$ 6;

5a. **Suffix-based Features**: Equal to final string of characters ($1 \leq$ n) in unstemmed word; Computed for n $\leq$ 6;

5b. **Suffix-based Features (skip character)**: Equal to final string of characters ($1 \leq$ n) in unstemmed word, skipping the mth character. Computed for n $\leq$ 6, m $\leq$ 6;

6a. **Previous Class Feature (NE class only)**: Equal to the value of named entity class (e.g. PERSON, ORGANIZATION) assigned to a previous word. Computed for Word(n-i), $1 \leq$ i $\leq$ 6;

6b. **Previous Class Feature (NE class + IOB annotation)**: Equal to the value of a named entity class, including the IOB annotation (e.g. I-PERSON, B-PERSON, O; Computed for Word(n-i), $1 \leq$ i $\leq$ 6;

7. **Morphological Features**: Set of boolean features derived (1) from the presence or absence of a bound morpheme or clitic (as detected using *light8* stemming), (2) the presence of the definite determiner \RLal- '*the*', or (3) a numeral.

Figure 7: Features used in CICEROARABIC

or a *B-type* token (if it considered the token to be the beginning of a new named entity expression) before moving on to the final token لَدن '*Laden*'.

Internal		Beginning
لَدن$_I$ − *person*	بن$_I$ − *person*	أُسَمة$_B$ − *person*
Laden$_I$ − *person*	*bin*$_I$ − *person*	*Usama*$_B$ − *person*
'Usama bin Laden'		

Table 2: Entity Boundary Detection

5. Evaluation

When trained on a randomly selected sample of 600,000 words from our training corpus, CICEROARABIC correctly recognizes 18 different types of named entities with an average of over 85% F. Performance for each of the individual named entity types found in the CICEROARABIC named entity ontology is presented in Table 3.

While performance varied significantly ($p<0.05$) across types, CICEROARABIC approached or exceeded 90% F for four different named entity types: (1) dates (97.87%), (2) time quantities (90.17%), (3) numbers (89.98%) and (4) political locations (89.27%).

Figure 8 presents an ablation study comparing the relative impact of four different classes of features used in CICEROARABIC. The complete set of features presented in Figure 7 were collapsed into four different categories: (1) *word-based* features, (2) *stem-based* features, and (3) *affix* features (including the two variants of prefix-based and suffix-based features).

In order to evaluate the effect of the number of classification outcomes on the performance of the system, we evaluated CICEROARABIC on two other named entity classification

	Entity Name	Abbr.	Precision	Recall	Fβ1
Names	Political Location	GPE	93.89	85.08	89.27
	Geographic Location	LOC	94.00	78.33	85.45
	Person	PER	85.02	76.00	80.26
	Organization	ORG	80.08	66.35	72.57
Numeric	Number	NBR	91.21	88.78	89.98
	Percent	PCT	93.75	75.00	83.33
	Temperature	TEM	96.52	71.33	82.03
	Age	AGE	94.74	58.06	72.00
	Monetary Amount	MON	58.00	67.44	62.37
	Unit	UNT	66.67	56.00	60.87
Time	Date	DAT	98.20	97.53	97.87
	Time Quantity	TQY	86.93	93.66	90.17
	Time	TIM	78.57	84.62	81.48
Misc	Weapon	WEA	93.50	84.55	88.80
	Person Count	PCN	71.43	84.07	77.24
	Non-Person Count	OCN	74.47	67.31	70.71
	Vehicle	VEH	80.00	28.57	42.11
	Facility	FAC	75.00	22.22	34.29
Total (18 Classes)			88.77	82.49	85.51

Table 3: Performance by Individual Classes

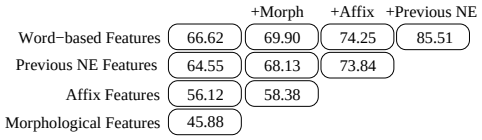

Figure 8: Comparison of Feature Performance

systems: (1) a modified version of the CoNLL set of 3 entity types and (2) the MUC-7 set of eight entity types. In order to compare these systems directly, the MUC-7 LOCATION type was considered to be equal to the union of the CICEROARABIC *political location* and *geographic location* types, while the MUC-7 *relative time* type was considered to be roughly equivalent to the CICEROARABIC *time quantity* type. Since the CoNLL *Miscellaneous* class (MISC) (Tjong Kim Sang and De Meulder, 2003) was not analogous to any of the categories in our annotations, it was not considered in this test. F-measure for each classification system is provided in Table 4. While CICEROARABIC's performance did increase with the size of the training corpus for each of the three different classification systems, performance did not significantly ($p<0.05$) degrade when the system was moved from the smaller CoNLL and MUC-7 classification systems to the much larger CICEROARABIC set of 18 classes.

Training Data (# of Words)	**CoNLL** (3 Types)	**MUC-7** (8 Types)	**CiceroArabic** (18 Types)
100K	78.62	82.37	82.04
300K	79.93	83.50	83.04
600K	82.79	85.54	85.51

Table 4: 3 Different Classification Systems

We have found that CICEROARABIC's performance on the 4 "core" entity types – *person*, *organization*, *political location* (GPE), and *geographic location* (LOC) – remains remarkably stable, regardless of the total number of named entities it must classify. Table 5 details our system's performance for the 4 core NE types under 8 different NE classification systems. In order to produce each of these 8 permutations, we grouped CICEROARABIC's 18 named entity types into four classes of mention types: (1) *named entities* (NAM), (2) *numeric expressions* (NUM), (3) *time expressions* (TIM), and (4) a *miscellaneous* (MISC) class that included both the set of quantity types (e.g. *person count*, *other count*) and the set of artifact types (*weapon*, *vehicle*,

etc.). No one permutation resulted in the best score for multiple core types: while the system correctly classified GPEs with the highest accuracy when only NAM types were considered, classification of LOCs was significantly improved when all four mention types were considered. While the average performance for the 4 core types only varied 0.37% across the 8 permutations, the best average performance was achieved when the MISC category – on average, the worst performing of the four mention types – was excluded.

Mention Types	F-Measure				
	PER	ORG	GPE	LOC	*Average*
+NAM	79.65	71.88	90.54	80.36	82.79
+NAM,+NUM	79.76	72.13	89.55	83.02	82.52
+NAM,+TIM	79.08	72.31	90.07	80.77	82.57
+NAM,+MISC	80.24	72.13	89.21	84.40	82.54
+NAM,+NUM,+TIM	79.46	73.47	89.84	81.13	82.83
+NAM,+NUM,+MISC	80.55	72.29	88.95	82.24	82.52
+NAM,+TIM,+MISC	79.83	70.92	89.86	82.24	82.46
+NAM,+NUM,+TIM,+MISC	80.26	72.57	89.27	85.45	82.72

Table 5: Performance across Mention Types

With the exception of the MISC category, we found that the system made most of its classification errors within a mention type category. Table 6 presents a confusion matrix for the 4 mention types considered above.

		Predicted Mention Type				
		NAM	NUM	TIM	MISC	TOTAL
Actual	NAM	453	52	0	179	684
	NUM	142	78	26	41	287
	TIM	25	1	401	212	639
	MISC	43	15	79	7	144
	TOTAL	663	146	506	439	1754

Table 6: Confusion Matrix

For three of the four mention types, CICEROARABIC was more likely to assign an incorrect entity type from the appropriate mention type category than to assign an entity type from an inappropriate mention type. For example, 68% (453/663) of examples incorrectly labeled as one of the 4 NAM were actually instances of the NAM mention type; this trend was repeated for NUM (53%) and TIM (79%) as well. This trend was not observed with the MISC category, however: in this case, only 7 out of 439 incorrectly labeled examples (1.6%) were actually found to be other instances of the MISC mention type category.

6. Conclusions

In this paper, we have presented details of a new NER system for Modern Standard Arabic, known as CICEROARABIC, which is capable of recognizing a total of 18 named entity types with over 85% F. As part of this work, we defined a new set of NE tagging guidelines and applied them to create a new 800,000-word Arabic corpus annotated with named entity information. While these guidelines only account for a small number of the possible named entity classes that could be potentially recognized in Arabic, we have demonstrated that this selection of classes enabled our system to recognize a larger number of entities without compromising performance on core entity types.

7. Acknowledgments

This material is based upon work funded in whole or in part by the U.S. Government and any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

8. References

Douglas Appelt, Jerry Hobbs, John Bear, David Israel, Megumi Kameyama, Andy Kehler, David Martin, Karen Meyers, and Mabry Tyson. 1995. Sri international fastus system: Muc-6 test results and analysis. In *In Proceedings of the Sixth Message Understanding Conf. (MUC-6)*. Morgan Kaufman.

D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: A high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*.

Nancy Chinchor, 1997. *MUC-7 Named Entity Task Defi nition*.

S. Cucerzan and D. Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL 2004*.

Jan Hajic, Otakar Smrz, Petr Zemanek, Petr Pajas, Jan Snaidauf, Emanuel Beska, Jakub Kracmar, and Kamila Hassanova. 2004. *Prague Arabic Dependency Bank 1.0*. Number 1.0 LDC2004T23.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunsecu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2001. The role of lexico-semantic feedback in open-domain textual question-answering. In *ACL01*.

Dan Klein, Joseph Smarr, Huy Nguyen, and Chris Manning. 2003. Named entity recognition with character-level models. In *Conference on Natural Language Learning '03*.

L. Larkey, L. Ballesteros, and M. Connell. 2002. Improving stemming for arabic information retrieval: Light stemming and cooccurrence analysis. In *SIGIR*.

Linguistic Data Consortium, 2005. *ACE (Automatic Content Extraction) Arabic Annotation Guidelines for Entities*.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR International Conference on Arabic Language Resources and Tools*.

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*.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the Thirteenth ACM conference on Information and knowledge management*.

Mihai Surdeanu and Sanda Harabagiu. 2002. Infrastructure for open-domain information extraction. In *Proceedings of the Human Language Technology Conference*.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*.