A Survey of Arabic Named Entity Recognition and Classification

Khaled Shaalan*
School of Informatics, University of Edinburgh, UK
The British University in Dubai, UAE

As more and more Arabic textual information becomes available through the Web in homes and businesses, via Internet and Intranet services, there is an urgent need for technologies and tools to process the relevant information. Named Entity Recognition (NER) is an Information Extraction task that has become an integral part of many other Natural Language Processing (NLP) tasks, such as Machine Translation and Information Retrieval. Arabic NER has begun to receive attention in recent years. The characteristics and peculiarities of Arabic, a member of the Semitic languages family, make dealing with NER a challenge. The performance of an Arabic NER component affects the overall performance of the NLP system in a positive manner. This article attempts to describe and detail the recent increase in interest and progress made in Arabic NER research. The importance of the NER task is demonstrated, the main characteristics of the Arabic language are highlighted, and the aspects of standardization in annotating named entities are illustrated. Moreover, the different Arabic linguistic resources are presented and the approaches used in Arabic NER field are explained. The features of common tools used in Arabic NER are described, and standard evaluation metrics are illustrated. In addition, a review of the state of the art of Arabic NER research is discussed. Finally, we present our conclusions. Throughout the presentation, illustrative examples are used for clarification.

1. Introduction

In the 1990s, in particular at the Message Understanding Conferences, Named Entity Recognition (NER) was first introduced as an information extraction task and deemed important by the research community. In NER, the expression "named entity" (NE) covers not only proper names but also includes temporal expressions and some numerical expressions such as monetary amounts and other types of units. Proper names include three classic specializations (referred to as **types** or **classes** in the literature): persons, locations, and organizations. For example, in the sentence *Ahmed Khaled, CEO of Arabisoft Company in Egypt*, *Ahmed Khaled*, *Arabisoft Company*, and *Egypt* would be identified as references to a person, an organization, and a location, respectively. A type can in turn be divided into subtypes (Sekine, Sudo, and Nobata 2002), possibly forming an entity type hierarchy (Pappu 2009). For example, locations might be divided into

* The British University in Dubai (BUiD), P.O. Box 345015, Dubai, UAE.
E-mail: khaled.shaalan@buid.ac.ae.

Submission received: 12 September 2012; revised submission received: 12 March 2013; accepted for publication: 17 July 2013.

doi:10.1162/COLI_a_00178

multiple fine-grained locations, such as city, state, and country. For specific needs other types might be introduced, such as e-mail address, phone number, book ISBN, filename, and so on.

A good portion of NER research is devoted to the study of English, due to its significance as a dominant language that is used internationally for communications, science, information technology, business, seafaring, aviation, entertainment, and diplomacy. This has limited the diversity of text genre and domain factors from other languages that are usually considered when developing NER for these fields. For instance, as most scientific studies are conducted in English in almost all Arabic-speaking countries, there is no urgency to investigate Arabic NER for areas such as bioinformatics, drug, or chemical named entities.

NER can be defined as the task that attempts to locate, extract, and automatically classify named entities into predefined classes or types in open-domain and unstructured texts, such as newspaper articles (Nadeau and Sekine 2007). One obvious reason for the importance of named entities is their pervasiveness, which is evidenced by the high frequency, including occurrence and co-occurrence, of named entities in corpora (cf. Saravanan et al. 2012). Arabic is a language of rich morphology and syntax. Its characteristics and peculiarities make dealing with it a challenge (Farghaly and Shaalan 2009). The last decade has shown a growing interest in addressing challenges that underlie the development of a productive and robust Arabic NER system (Al-Jumaily et al. 2012; Oudah and Shaalan 2012).

This article investigates the progress in Arabic NER research. The survey by Nadeau and Sekine (2007) presents background on much of the work on NER for a variety of languages and myriad machine learning (ML) techniques. To the best of our knowledge, Arabic NER and classification have not yet been surveyed extensively, which has motivated us to conduct this survey.

The survey is structured as follows. Section 2 provides background information relevant for working with Arabic NER. Section 3 presents some aspects of the Arabic language that will allow the reader to appreciate the difficulties associated with Arabic NER. Section 4 briefly introduces the standard tag sets commonly used to annotate named entities. Section 5 describes the Arabic NER language-specific resources that are involved in the NER task. Section 6 gives a brief description of approaches used in Arabic NER. Section 7 discusses feature selection, which is a critical factor for achieving better performance for NER systems. Section 8 presents various tools that have been used in building Arabic NER systems and Section 9 illustrates evaluation techniques for NER systems. Section 10 presents the state-of-the-art in Arabic NER research. Finally, the concluding remarks are presented in Section 11.

2. Background

2.1 Entity Tracking

The task of identifying named entities must be distinguished from entity tracking, which involves identifying mentions, relations, and the co-references that may exist between them. In this regard, a NE may contain only one mention such as a person name (e.g., *Mohammed Morsi*), but when a pronoun is used to refer to the same person, it is considered another mention of that entity. Moreover, a nominal (e.g., *president*) can also be used as a mention to refer to the same NE (cf. Zitouni et al. 2005). It should be noted that the richness of Arabic morphology allows two mentions to appear in one word (e.g., رئيسنا *our president*, *president-our*), where a pronominal (نا, *our*) can appear as a suffix pronoun to a nominal (e.g., *president* رئيس). A co-reference exists when a group of mentions refers to the same entity. For example, in the sentence *The [Egyptian President], [Mohammad Morsi], as the [chair of the 15th Non-Aligned Movement summit] declared opening of the 16th summit*, there are three mentions that refer to the same person. Mentions also include aliases such as *Abu Ammar*, which refers to the same entity as *Yasser Arafat*.

An entity relation may be established between two or more NEs, such as a person, an organization, a location, or a specific time. Relationships between NEs can be binary, such as person-affiliation or organization-location, or may involve more entities; for example, *[a person] is in [a place] at [a specific time]*. The entity relation is usually expressed in a predicate form and is used to establish relations such as whether two persons were working at the same organization at the same time (Ben Hamadou, Odile, and Héla 2010a).

In summary, it is important to direct attention to the choice of the recognition unit (i.e., real world NE, mention, co-reference, or relation), because mention detection, co-reference resolution, and relation extraction are considered more difficult than the traditional NER task due to the complexity incurred by extracting non-named mentions, grouping mentions into entities, and deriving semantic relations among entities.

2.2 The Broader Role of NER

The implications of research in NER for NLP more generally are too obvious to enumerate. Examples of applications for which NER is useful are shown in this section.

Information Retrieval. This is the task of identifying and retrieving relevant documents from a set of data according to an input query. A study by Guo et al. (2009) has indicated that about 71% of the queries in search engines contain NEs. Information Retrieval can benefit from NER in two phases (Benajiba, Diab, and Rosso 2009a): firstly, recognizing the NEs within the query; and secondly, recognizing the NEs within the searched documents, and then extracting the relevant documents taking into account their classified NEs and how they are related to the query. For example, the word الجزيرة (*Aljazeera*) can be recognized as an organization name or a noun corresponding to the word island; the correct classification will facilitate extracting relevant documents.

Question Answering. This is very similar to Information Retrieval but with more sophisticated results. A Question Answering system takes questions as input and gives in return concise and precise answers (Ezzeldin and Shaheen 2012). The NER task can be utilized in the phase of analyzing the question so as to recognize the NEs within the question that will help later in identifying the relevant documents and constructing the answer from relevant passages (Mollá, van Zaanen, and Smith 2006; Badawy, Shaheen, and Hamadene 2011; Lahsen, Bouzoubaa, and Rosso 2012). For instance, the NE الشرق الأوسط (*Middle East*) may be classified as an organization name (e.g., a newspaper) or as a location name according to the context. Hence, the correct classification for the NE will help to target the relevant group of documents that answer the input query. Moreover, Question Answering systems could benefit substantially from NER, because the answer to many factoid questions involve NEs (Trigui et al. 2012) (e.g., answers to *who* (ماهي/من هو) questions usually involve persons or organizations, *where* (أين)

questions involve locations, and *when* (متى) questions involve temporal expressions) (Brini et al. 2009).

Machine Translation. This is the task of automatically translating a text from one natural language into another. NEs need special attention in order to decide which parts of an NE should be meaning-translated and which parts should be phoneme-transliterated (Al-Onaizan and Knight 2002b; Hassan and Sorensen 2005). Usually this depends on the type of the NE (Chen, Yang, and Lin 2003). For example, personal names tend to be transliterated.[1] For a location name, the name part and the category part (e.g., *mountains*) are usually transliterated and translated, respectively. Organization names are completely different in that most of the constituents are translated (e.g., *United Nations*). The quality of the NER system plays a significant role in determining the overall quality of the machine translation system, and hence, NE translation is critical for most multilingual application systems (Babych and Hartley 2003; Ben Hamadou, Odile, and Héla 2010b; Steinberger 2012). In addition, NE translation is very important for other applications such as cross-lingual information retrieval for extracting newly introduced NEs from the Web and news documents and regularly updating the list of NE translation pairs (Hassan, Fahmy, and Hassan 2007).

Text Clustering. Search results clustering may exploit NER by ranking the resulting clusters based on the ratio of entities each cluster contains (Benajiba, Diab, and Rosso 2009a). This enhances the process of analyzing the nature of each cluster and also improves the clustering approach in terms of selected features. For example, time expressions along with location NEs can be utilized as factors that will give an indication of when and where the events mentioned in a cluster of documents have occurred.

Navigation Systems. These systems, which facilitate navigation using digital maps, now play significant roles in our lives. They provide directions, information about nearby places possibly linked with other on-line resources, and traffic conditions. In these systems, points of interest (also known as waypoints) are NEs that are stored in a database with their geographic coordinates (Kim, Kim, and Cho 2012). They refer to areas of interest that are typically of significance to, among others, tourists, visitors, and rescuers, allowing the location of places such as parking areas, shops, hospitals, restaurants, universities, schools, landmarks, and so on.

3. Linguistic Issues and Challenges

Arabic is a highly inflected language, with a rich morphology and complex syntax (Al-Sughaiyer and Al-Kharashi 2004; Ryding 2005). Current Arabic NLP research efforts cannot cope with the massive growth of Arabic data on the Internet and the heightened need for accurate and robust processing tools (Abdul-Mageed, Diab, and Korayem 2011). NER is considered one of the building blocks of Arabic NLP tools and applications. Though significant progress has been achieved in Arabic NER research in the last decade, the task remains challenging due to the following

[1] Transliteration is the task of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. It unambiguously represents the graphemes, rather than the phonemes, of the NE. Transliteration between languages that use similar alphabets and sound systems is very simple. However, transliterating NEs between Arabic and English is a non-trivial task, mainly due to the differences in their sound and writing systems (Al-Onaizan and Knight 2002a).

features of the Arabic language; opportunities for improved performance are still available.

3.1 Arabic Script

The Arabic language relies on the Arabic script, which is also used in writing other languages such as Persian, Urdu, Kurdish, and Pashto (Habash 2010). Some researchers have developed Arabic computational tools and resources based on Romanized[2] or transliterated[3] Arabic text rather than genuine Arabic script (e.g., Buckwalter's Arabic Morphological Analyzer [Buckwalter 2002], the CJK lexical resources [Halpern 2009], and Arabic NER systems [Bidhend, Minaei-Bidgoli, and Jouzi 2012; Zayed and El-Beltagy 2012]), either because these formats are more familiar to non-native Arabic speakers or because of limitations in the Arabic script encoding imposed by the development environment. This approach should disappear over time with the rapidly growing quantity of Arabic script-based Web content and new technologies that support multiple encodings.

3.2 Language in Use

With regard to language usage, Arabic can be classified into three types (Elgibali 2005): Classical Arabic (CA), Modern Standard Arabic (MSA), and Colloquial Arabic Dialects (Abdel Monem et al. 2008; Habash 2010; Korayem, Crandall, and Abdul-Mageed 2012). As far as Arabic NE is concerned, it is important to know the difference between these various uses of the language. CA is the formal version that has been used continuously for over 1,500 years as the language of Islam, used by Muslims in their daily prayers. Most Arabic religious texts are written in CA. In this context, person name recognition is of particular interest in order to identify and verify the correctness of citations (Zaraket and Makhlouta 2012) (a sequence of hadith narrators referencing each other who provide narrations related to the Prophet Mohammed based on known truthful and untruthful relaters). The importance of verification is that the authenticity of a hadith needs to be established before his narration is used in jurisprudence, and this depends on the credibility of the narrators. Furthermore, many historical Arabic manuscripts are handwritten in CA (or Arabic calligraphy); when they are digitized and converted to text, Arabic NE will become important.

MSA is the language of today's Arabic newspapers, magazines, periodicals, letters, modern writers, and education. MSA is one of the six official languages of the United Nations used in meetings and official UN documents. Most Arabic NLP, including NER research projects, is focused on MSA. The main difference between MSA and CA lies in the vocabulary, including NEs, and the orthography of conventional written Arabic (Farber et al. 2008): MSA does not require the inclusion of short vowels. Moreover, MSA reflects the needs of contemporary expression, whereas CA reflects the needs of older styles. For example, the Arabic NEs in rare documents and old manuscripts that refer

[2] Transliteration from Arabic to languages using the Latin alphabet is called **Romanization**.

[3] In a multilingual context, transliteration of NEs would differ depending on the target language (Pouliquenet et al. 2005). For example, the Arabic name مصطفى could be transliterated into English as *Mustafa* or *Moustapha*, while a likely French transliteration would be *Moustafa* or *Moustapha*.

to places, jobs, or organizations are different from the corresponding NEs in modern documents.

Colloquial Arabic is the spoken Arabic used by Arabs in their informal daily communication; it is not taught in schools due to its irregularity. Unlike the widespread use of MSA across all Arab countries, colloquial Arabic is a regional variant that differs not only among Arab countries, but also across regions in the same country. Written Colloquial Arabic is presently used mainly in social media communication. For comparison, a person name in either CA or MSA could be expressed in Arabic dialect by more than one form; for example, عبد القادر (*Abd Al-Kader*) versus عبد الجادر (*Abd Al-Gader*) or عبد الآدر (*Abd Al-Aader*). Salloum and Habash (2012) presented a universal machine translation pre-processing approach that has the ability to produce MSA paraphrases of dialectal input. In this way, available MSA tools can also be used to process Colloquial Arabic text, as most of the Arabic NER systems are developed to support MSA.

3.3 Lack of Capitalization

Unlike languages like English that use the Latin script, where most NEs begin with a capital letter, capitalization is not a distinguishing orthographic feature of Arabic script for recognizing NEs such as proper names, acronyms, and abbreviations (Farber et al. 2008). The ambiguity caused by the absence of this feature is further increased by the fact that most Arabic proper nouns (NEs) are indistinguishable from forms that are common nouns and adjectives (non-NEs). Thus, an approach relying only on looking up entries in proper noun dictionaries would not be an appropriate way to tackle this problem, as ambiguous tokens/words that fall in this category are more likely to be used as non-proper nouns in text (Algahtani 2011). For example, the Arabic proper name أشرف (*Ashraf*) can be used in a sentence as a given name, an inflected verb (*he-supervised*), and a superlative (*the-most-honorable*) (Mesfar 2007). An NE is usually found in a context, namely, with trigger and cue words to the left and/or right of the NE. Therefore, it is common to resolve this type of ambiguity by analyzing the context surrounding the NE. However, this might require deeper analysis of the NE's context. As an example, consider the nominal sentence مسقط رأسه بجدة, whose literal meaning might be *the falling of his head in grandfather/Jeddah*. The correct analysis of the trigger constituent مسقط رأسه as a multiword expression denoting *place of birth* leads to the recognition of the following noun as a location name.

3.4 Agglutination

The agglutinative nature of Arabic results in many different patterns that create many lexical variations. Each word may consist of one or more prefixes, a stem or root, and one or more suffixes in different combinations, resulting in a very systematic but complicated morphology. Clitics, which in other languages such as English would be treated as separate words, agglutinate to words. Arabic has a set of clitics that are attached to an NE, including conjunctions such as و (Waw, *and*) and ف (*if ... then*) and prepositions such as ل (Laam, *for/to*), ك (k, *as*), and ب (baa, *by/with*), or a combination of both, as in ول (Waw-Laam, *and-for*). NER relies on the words forming the NE and the context in which it appears. Both the words and the contexts may appear in different inflected forms. In order to address data sparseness issues without requiring massive training corpora, these bound morphemes should undergo morphological preprocessing. One solution is to omit all the affixes and keep only the root morpheme (Grefenstette, Semmar, and Elkateb-Gara 2005; Farber et al. 2008; Alkharashi 2009). For example, the analysis of the word وبمصر (*and by Egypt*, and-by-Egypt) yields مصر (*Egypt*) as a location name. Another solution is to perform text segmentation and insert a delimiter between constituent morphemes, thus preventing loss of contextual information (Benajiba and Rosso 2007). This information is more convenient for NLP tasks that need to process these morphemes. As an example that shows an occurrence of both prefix and suffix morphemes, consider the trigger word وعاصمتها (*and its capital*, and-capital-its), which is segmented into three parts—a conjunction, and both a nominal and a pronominal mention—separated by a space character: و عاصمة ها (*and capital its*).

3.5 Optional Short Vowels

Arabic text contains diacritics representing most vowels that affect the phonetic representation and give different meaning to the same lexical form.[4] Nowadays, the modern version of Arabic is written without diacritics, creating a one-to-many, unvocalized-to-vocalized, ambiguity (Alkharashi 2009), which gives mutually incompatible morphological analyses for the same surface form. As such, most Arabic texts that appear in the media (whether in printed documents or digitized format) are undiacritized. This is comprehensible for native Arabic speakers, but not for a computational system. The simplification made by ignoring such diacritics had led to structural and lexical types of ambiguity because different diacritics represent different meanings. These ambiguities can only be resolved by contextual information and an adequate knowledge of the language (Benajiba, Diab, and Rosso 2009a). For instance, قطر may refer to the country name *Qatar* (a location NE) if transliterated as *qatar*, the literal meaning of *country* (a trigger word for location NEs), or *radius* (a trigger word for measure NEs) if transliterated as *qutr*, or the literal meaning of *distill* if transliterated as *qat~ar*. Unfortunately, this solution might not work if the contextual information is itself ambiguous due to non-vocalization (Mesfar 2007). To consider another example, the likely vocalizations of the unvoweled form مؤسسة might lead to trigger words that denote two different NE types (e.g., مُؤَسَّسَة [*a foundation/corporation*], internal evidence of a constituent of an organization name; and مُؤَسِسَة [*a founder*], a trigger word for personal names).

3.6 Inherent Ambiguity in Named Entities

Arabic, like other languages, faces the problem of ambiguity between two or more NEs. For example consider the following text: احمد اباد رحب بالفائزين (*Ahmed Abad welcomed the winners*). In this example, احمد اباد (*Ahmed Abad*) is both a person name and a location name, thereby giving rise to a conflict situation, where the same NE is tagged as two different NE types. Heuristic techniques for resolving ambiguities by cross-recognizing NE types are suggested. One heuristic technique, proposed by Shaalan and Raza (2009), uses heuristic rules for preferring one NE type over the other.

[4] A diacritic in Arabic is a small mark placed either above or under a letter to indicate what short vowel will follow that letter. Long vowels are usually indicated by one of three designated letters.

Another technique, proposed by Benajiba, Diab, and Rosso (2008b), favors the NE type for which the classifier achieves the highest precision.

3.7 Lack of Uniformity in Writing Styles

Arabic has a high level of transcriptional ambiguity: An NE can be transliterated in a multitude of ways (Shaalan and Raza 2007). This multiplicity arises from both differences among Arabic writers and ambiguous transcription schemes (Halpern 2009). The lack of standardization is significant and leads to many variants of the same word that are spelled differently but still correspond to the same word with the same meaning, creating a many-to-one, variants-to-well-formed, ambiguity. For example, transcribing (also known as "Arabizing") an NE such as *the city of Washington* into Arabic NE produces variants such as واشنجطن, واشنطن, واشنغطن, وشنطن. One reason for this is that Arabic has more speech sounds than Western European languages, which can ambiguously or erroneously lead to an NE having more variants. One solution is to retain all versions of the name variants with a possibility of linking them together. Another solution is to normalize each occurrence of the variant to a canonical form (Pouliquen et al. 2005); this requires a mechanism (such as string distance calculation) for name variant matching between a name variant and its normalized representation (Refaat and Madkour 2009; Steinberger 2012).

3.8 Systematic Spelling Mistakes

Typographic errors are frequently made by Arabic writers with regard to certain characters (Shaalan et al. 2012). This is due to either a character similarity or inherent disagreement about the characters, which often leads to orthographical confusion (El Kholy and Habash 2010; Habash 2010; Al-Jumaily et al. 2012). The former category includes the character *Ta-Marbuta* (ة), literally 'tied Ta', which is a special morphological marker typically marking a feminine ending; this is carelessly written interchangeably with Ha (ه). Ta-Marbuta is a hybrid character merging the form of the characters *Ha* (ه) and *Ta* (ت). The latter category includes the *Hamza-Alif* letter variants that are often reductively normalized by brute force replacement with a bare Alif. Some computational linguists avoid writing the Hamza (especially with stem-initial Alifs), viewing this as a Hamza restoration problem that is part of the Arabic diacritization problem. As an example that combines both types of errors, consider الجامعة الأسلامية بجدة (*The Islamic University in Jeddah*), which might be written with both typographical variants as الجامعه الاسلاميه بجده. An edit-distance technique can be used to resolve the spelling variant problem. It should be noted that not all systematic spelling mistakes can be handled in this way. For example, consider the difference between بالجامعة (*and by/with the university*) and بلاجامعة (*without a university*). It is difficult to determine whether or not this mistake is due to the transposition of the two characters ا (*Alif*) and ل (*Lam*), where the prefix ال (means *the*) whereas the prefix لا (means *no*). The latter variation also shows another orthographic problem: Arabic "run-on" words, or free concatenation of words, when the word immediately preceding ends with a non-connector letter, such as ا (Alif), د (Dal), ذ (Dhal), ر (Ra), ز (za), و (waw), and so forth. For example, the following phrase shows a fully concatenated person NE and its surrounding context: الدكتورمحمدوزيرالخارجية (*Dr-Mohammed-the-Minister-of-Foreign-Affairs*). This is comprehensible by most readers but not by a computational system that needs to work on segmented words.

3.9 Lack of Resources

Large collections of tagged documents (corpora) as well as gazetteers (predefined lists of typed NEs) are excellent sources that we can rely upon when implementing and testing the performance of an Arabic NER system. For these linguistic resources to be useful, they should include unbiased distribution and representative numbers of NEs that do not suffer from sparseness. Unfortunately, the available Arabic resources for NER research often have limited capacity and/or coverage (Abouenour, Bouzoubaa, and Rosso 2010). Moreover, it is expensive to create or license these important Arabic NER resources (Huang et al. 2004; Bies, DiPersio, and Maamouri 2012). For these reasons, researchers often rely on their own corpora, which require human annotation and verification. Few of these corpora have been made freely and publicly available for research purposes (Benajiba, Rosso, and Benedí Ruiz 2007; Benajiba and Rosso 2007; Mohit et al. 2012), whereas others are available but under license agreements (Strassel, Mitchell, and Huang 2003; Mostefa et al. 2009).

4. Named Entity Tag Set

Tagging, also known as labeling, is the task of assigning a contextually appropriate tag (label) to every NE in the text. The sequence of words that is annotated with the same tag is considered a single multiword NE. The tag set used to tag NEs may differ according to user requirements. For example, Nezda et al. (2006) used an extended set of 18 different NE classes. Mohit et al. (2012)'s research adopted a very flexible scheme that allows annotators more freedom in defining entity types. In this research, entity types were not predetermined and category matches between annotators were determined by post hoc analysis.

In the literature, there are three standard general-purpose tag sets that have been used to annotate Arabic linguistic resources in the field of NER research. These tag sets may be used as a basis for annotating linguistic resources and system outputs.

The 6th Message Understanding Conference (MUC-6):[5] This conference can be considered as the initiator of the NER task. NEs are classified into three main tag elements: ENAMEX (i.e., person name, location, and organization), NUMEX (i.e., money and percentage [numerical] expressions), and TIMEX (i.e., time and date expressions). Each tag element is categorized via the TYPE attribute. Most researchers adopt this tag set. For example, a NER system producing MUC-style output might tag the sentence إشترى خالد ٣٠٠ سهم من شركة آبل فى ٢٠١٢ (*Khaled bought 300 shares of Apple Corp.*) as illustrated in Table 1.

The Conference on Computational Natural Language Learning (CoNLL): As an outcome of CoNLL2002[6] and CoNLL2003, four categories of NEs were defined: person name, location, organization, and miscellaneous. CoNLL follows the IOB format to tag chunks of text representing NEs in a data set (Benajiba, Rosso, and Benedí Ruiz 2007). The CoNLL annotations are formulated as a word-based classification problem, where each word in the text is assigned a tag, indicating whether it is the beginning (B) of a specific NE, inside (I) a specific NE, or (O) outside any NE. IOB notation is used when NEs are not nested and therefore do not overlap. For example, a NER system producing CoNLL-style output might tag the sentence

[5] http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html.

[6] http://ifarm.nl/signll/conll/.

Table 1
Example of MUC tagging.

⟨/ENAMEX⟩	خالد	⟨ENAMEX TYPE=PERSON⟩	إشتري
سهم من	⟨/NUMEX⟩	300	⟨NUMEX TYPE=CARDINAL⟩
في	⟨/ENAMEX⟩	شركة أبل	⟨ENAMEX TYPE=ORGANIZATION⟩
	⟨/TIMEX⟩	2012	⟨TIMEX TYPE=DATE ⟩

Table 2
Example of CoNLL tagging.

Arabic	English Trans.	Tag
فرانكفورت	*Frankfurt*	B-LOC
أعلن	*said*	O
اتحاد	*Association*	B-ORG
صناعة	*Industry*	I-ORG
السيارات	*Auto*	I-ORG
في	*in*	O
المانيا	*Germany*	B-LOC

Table 3
Example of ACE tagging.

في العام الماضي	⟨/GPE⟩	لبنان	⟨/GPE⟩	⟨/PER⟩	حسين	⟨PER⟩	زار الملك

فرانكفورت ، أعلن اتحاد صناعة السيارات في ألمانيا (*Frankfurt, Auto Industry Association in Germany said*) as illustrated in Table 2.

BILOU (Ratinov and Roth 2009) was also suggested as an efficient alternative to the BIO format. It is used to identify the beginning, the inside, and the last tokens of multi-token chunks as well as unit-length chunks. Experimental results indicate that BILOU representation of text chunks significantly outperforms the BIO format.

The Automatic Content Extraction (ACE) program: Arabic resources for Information Extraction have been developed as part of the ACE program. According to the ACE 2003 tag elements,[7] four categories are defined: person name, facility, organization, and geographical and political entities (GPE). Later in ACE 2004 and 2005, two categories were added to this tag set: vehicles and weapons. For example, a NER system producing ACE-style output might tag the sentence زار الملك حسين لبنان في العام الماضي (*King Hussein visited Lebanon last year*) (Habash 2010) as illustrated in Table 3.

[7] The ACE tag sets for English, Arabic, and Chinese are available at `http://projects.ldc.upenn.edu/ace/data/`.

5. Arabic Linguistic Resources

The lack of digital linguistic resources creates a formidable obstacle when it comes to Arabic NLP in general and Arabic NER in particular. Investing in these resources is justified because it would lead to many benefits such as reusability, broad coverage, and frequency and distributional information, as well as a way of evaluating and comparing systems. Corpora and lexical resources are two main types of linguistic resources that are commonly used in NER.

5.1 Corpora

The corpus needed for NER is a sufficiently large annotated corpus where every NE has a type assigned to it. An important characteristic of a reliable corpus is that it should be well balanced in terms of the NE type distribution. A corpus can be genre independent/specific; domain independent/specific: and consist of texts in one natural language (a monolingual corpus), two natural languages (a bilingual, parallel, or comparable corpus), or more natural languages (a multilingual or crosslingual corpus). In Hassan, Fahmy, and Hassan (2007), a general framework is proposed for extracting NE translation pairs from both comparable and parallel corpora. Parallel corpora that are aligned on the sentence level have been used to tag one corpus based on the tagged information in the other corpus such that they can complement and improve each other (Benajiba et al. 2010; Burkett et al. 2010; Ma 2010). For example, Samy, Moreno, and Guirao's (2005) approach creates an NE aligned bilingual corpus that relies on the basic assumption that, given a pair of sentences where each one is the translation of the other, and given that in one sentence one or more NE were detected, then the corresponding aligned sentence should contain the same NE either translated or transliterated. As described, the approach is very effective because it involves Arabic, which is a case-insensitive language, and Spanish, which does have orthographical differences between names and non-names.

Experimental results of NLP research are more easily compared with each other when they rely on publicly available data sets or corpora. The frequent use of these corpora in the research community makes them standard data sets or corpora, serving as stable benchmark data for measuring ongoing progress and ranking systems according to their annotation capability. Some NER corpora are available to members of organizations under paid license agreements, for example, ACE[8](Strassel, Mitchell, and Huang 2003). Because they are not free, it is difficult for small research groups to access them. However, contributors from the Arabic NLP research community are striving to develop freely available Arabic NER corpora to alleviate this problem and help other researchers to exploit these resources, for example, ANERcorp[9](Benajiba, Rosso, and Benedí Ruiz 2007; Benajiba and Rosso 2007). Nonetheless, these efforts are still limited and focused around a small set of domains (Mohit et al. 2012). In most cases where researchers want to conduct further investigation by studying the impact of different parameters and new features of NER, therefore, they have found that it is indispensable to build their own corpora. In the literature, common and recent examples of Arabic corpora that have

[8] ACE corpora are available under license agreement from LDC (`http://www.ldc.upenn.edu`). A significant number of the data sets developed by LDC are Arabic language resources, making LDC the leading source for such materials (Bies, DiPersio, and Maamouri 2012).

[9] Available for free at `http://www1.ccls.columbia.edu/~ybenajiba/downloads.html`.

been used for Arabic NLP in general, and for Arabic NER and classification topics in particular, are:

- *ACE 2003 corpus*: This includes Broadcast News (BN) and Newswire (NW) genres. The total size is 55.29 KB and the number of NEs is 5,505.
- *ACE 2004 corpus*: This includes BN and NW from Arabic Tree Bank (ATB) genres. The total size is 154.12 KB and the number of NEs is 11,520.
- *ACE 2005 corpus*: This includes BN, NW, and Weblogs (WL) genres. The total size is 104.65 KB and the number of NEs is 10,218.
- *ANERcorp*: This includes NW genre. The corpus size is 174.76 KB and the number of NEs is 12,989.

5.2 Lexical Resources

Another primary linguistic resource is the **gazetteer**, which is a collection of predefined lists of typed entities; a gazetteer is also known as a dictionary or whitelist (Shaalan and Raza 2008). Gazetteers include names that have been identified beforehand and have been classified into NE types. When the acquisition of a gazetteer is fully automated, the number of NEs increases with the growth of the input linguistic resource or text used to create it. The contents of a gazetteer should be consistent and belong to only one type of NE. For example, a location gazetteer consists of names of continents, countries, cities, states, political regions, towns, and villages, and so on (Shaalan and Raza 2009). A gazetteer might include full or partial NEs; for example, a person NE could have separate gazetteers for first names (possibly distinguishing male names and female names), middle names, surnames, full forms, and even nicknames (Shaalan and Raza 2007; Higgins, McGrath, and Moretto 2010). A gazetteer entry provides internal evidence to fully or partially match a candidate NE in the input. Whenever a predefined NE that appears in the relevant gazetteer is detected in the input text, the NER system should recognize it directly as an NE of this type. Very large gazetteers are publicly available from the CJK Dictionary Institute[10] under license agreement in the form of Arabic person, organization, company, and location name databases. However, researchers who find these resources difficult to acquire build their own gazetteers from different resources such as the Web and from organizations (Benajiba and Rosso 2008; Shaalan and Raza 2009).

Some systems used a blacklist (Shaalan and Raza 2009) that allows for discarding of negative evidence. A filtering mechanism is used to reject incorrect matches. To see how this works, consider the following example: وزير الخارجية العراقى الامين العام (*The Iraqi Foreign Minister the Secretary-General*). The contextual information العراقي وزير الخارجية (*The Iraqi Foreign Minister*) indicates that the following words are a person name. However, in this example, the following words, الامين العام (*the Secretary-General*) do not constitute a valid person name; rather, they form an appositive which should be filtered out from the results.

Lexical triggers are also considered one of the important linguistic resources (Shaalan and Raza 2007). There are two kinds of lexical triggers that provide either internal or contextual evidence. The internal evidence lies within the NE itself, for

[10] See Arabic lexical resources at `http://www.cjk.org/cjk/arabic/arabsam.htm`.

example, شركة (*company*) is internal evidence of an organization NE. Contextual evidence is provided by the clues around the entities. They might be deduced from analysis of the most frequent left- and right-hand-side contexts. For example, the phrase دكتور محمد مرسي الرئيس المصري المنتخب حديثا (*Dr Mohammed Morsi the newly elected Egyptian president*) includes the preceding lexical trigger دكتور (*Dr*) and the following lexical triggers رئيس (*president*) and مصري (*Egyptian*) for the person NE محمد مرسي (*Mohammed Morsi*). Generally, lexical triggers provide clues that would indicate the presence or absence of NEs.

As far as the morphological properties are concerned, additional Arabic resources are needed to furnish information to NER systems, including lemmas, dictionaries, affix compatibility tables, and English glosses. For example, the English gloss, which is derived as a companion to some Arabic morphological analyzers, is used to check whether it starts with a capital letter, a key clue for an English NER. Its presence functions as a hint that suggests the presence of an Arabic NE. Benajiba, Rosso, and Benedí Ruiz (2007), among others, have used POS tags to improve NE boundary detection. Morphological information can be obtained from deep Arabic morphological analysis (Farber et al. 2008). However, leading and trailing character *n*-grams in surface word forms can also be used to handle affix attachment without the need for morphological analysis (Abdul-Hamid and Darwish 2010).

6. NER Approaches

A number of Arabic NER systems have been developed using primarily two approaches: the rule-based (linguistic-based) approach, notably the NERA system (Shaalan and Raza 2009); and the ML-based approach, notably ANERsys 2.0 (Benajiba, Rosso, and Benedí Ruiz 2007). Rule-based NER systems rely on handcrafted local grammatical rules written by linguists. Grammar rules make use of gazetteers and lexical triggers in the context in which the NEs appear. The main advantage of the rule-based NER systems is that they are based on a core of solid linguistic knowledge (Shaalan 2010). However, any maintenance or updates required for these systems is labor-intensive and time-consuming; the problem is compounded if the linguists with the required knowledge and background are not available. On the other hand, ML-based NER systems utilize learning algorithms that require large tagged data sets for training and testing (Hewavitharana and Vogel 2011). ML algorithms involve a selected set of features extracted from data sets annotated with NEs in order to generate statistical models for NE prediction. An advantage of the ML-based NER systems is that they are adaptable and updatable with minimal time and effort as long as sufficiently large data sets are available. Moreover, if we deal with an unrestricted domain, it is better to choose the ML approach, as it would be expensive both in terms of cost and time to acquire and/or derive rules and gazetteers. Recently, a hybrid Arabic NER approach that combines ML and rule-based approaches has resulted in significant improvement by exploiting the rule-based decisions of NEs as features used by the ML classifier (Abdallah, Shaalan, and Shoaib 2012; Oudah and Shaalan 2012). For a comprehensive survey of NER approaches more generally, see Nadeau and Sekine (2007).

Arabic morphology is relatively complex, so morphological information is needed in these approaches for identifying NEs. For example, consider the phrase أعلنت وزارة الداخلية المصرية (*The Ministry of Egyptian Interior announced*, announced the-ministry the-interior the-Egyptian). In this case, the rule or pattern that allows the recognizer to identify وزارة الداخلية المصرية (*The Ministry of Egyptian Interior*) as an organization name stipulates that if the NE is preceded directly by a verb trigger and is followed by a noun (internal evidence of an NE constituent), which in turn is followed by one or two specific adjectives, then the sequence of these two or three words should be tagged as an organization entity. For more precise identification of NEs, sometimes the adjective forms of nationality are also used in the recognition process (e.g., المصرية, *the-Egyptian.fem from Egypt*). Known organization NEs that are kept in the organization gazetteer can be used to improve the performance of the NER system. As such, the system is able to recognize وزارة الخارجية المصرية (*The Ministry of Egyptian Foreign Affairs*) in the short conjunction of organization NEs وزارتى الداخلية والخارجية المصرية (*Egyptian Ministries of Interior and Foreign Affairs*, Ministries.dual the-interior an̈d the-Foreign-Affairs Egyptian) by using the gazetteer entry for وزارة الداخلية المصرية (*The Ministry of Egyptian Interior*).

7. Feature Space of Arabic Named Entity Recognition

Features in NER are properties or characteristic attributes of words designed for consumption by a computational system. This process begins by transforming the set of words (tokens) to be categorized into a set of feature vectors that belong to a feature space, which is fed to the text classifier as input. The feature vector representation is an abstraction over the text, which usually characterizes each word by one or more Boolean or binary values (such as whether a word is capitalized), numerical values (word length), and nominal values (English gloss). The source of these values might be their appearance as surface features, a pre-processing step, surrounding items, or the characters that the word is composed of, or a combination of several features, or external knowledge (Oudah and Shaalan 2013).

In this section, we present the features most often used for the recognition and classification of Arabic NEs. We organize[11] them along the following different axes: word-level features, list lookup features, contextual features, and language-specific features. In the ML approach, the selection of the features to be taken into account by a classifier is a very critical issue and can significantly affect the performance of a system. Section 7.5 is dedicated to discussing the feature selection step.

7.1 Word-Level Features

Word-level features are related to the individual orthographic nature and structure of each word. Table 4 lists subcategories of these features. They specifically describe special markers and special characters, word length, corresponding English word case, and affix segments. Special markers are used to indicate an abbreviation (e.g., acronym or contraction) that might include internal periods, a hyphen, an ampersand, and so on. Word length is sometimes used to indicate the minimum length required in order for the word to be considered as an NE type. This feature capitalizes on the fact that short words are unlikely to be NEs.

Capitalization is a key feature of an English NER. Arabic is at a disadvantage in this regard because the script does not orthographically mark proper names in this way. However, many researchers (e.g., Benajiba, Diab, and Rosso 2008a; Mohit et al. 2012; Farber et al. 2008), have been able to derive the assumed capitalization from the

[11] In the literature, other ways used to classify features are linguistic-dependent versus independent features and contextual versus internal features.

Table 4
Word-level features.

Feature	Description
Special markers	A binary feature indicating the presence of punctuation marks and special characters in a word.
Word length	A binary feature indicating whether the length of the word is greater than a predefined threshold.
Capitalization	A binary feature indicating the existence of capitalization information on the gloss corresponding to the Arabic word.
Lexical	The surface features of a character *n*-gram up to a range of characters from 1 to *n* that indicate prefix and suffix attachment.

lexical correspondences between Arabic and English, based on the underlying bilingual lexicon of BAMA (Buckwalter 2002) that MADA exploits (Habash and Rambow 2005). The capitalization feature has been designed with this in mind. The insight is that if the translation begins with a capital letter then it is most probably an NE.

One of the major problems of the Arabic language is the large number of prefixes and suffixes that are attached to an inflected word. Lexical features are extracted via pattern matching rather than linguistic processing. Hence, in the literature they are considered language-independent features that capture the word prefix and suffix character sequences of length up to *n*. The sequences are matched from the leftmost (prefix) and rightmost (suffix) positions of the words. In Benajiba, Diab, and Rosso (2008b) and Abdul-Hamid and Darwish (2010), lexical features are represented by character *n*-grams of leading and trailing characters in a word, which can frequently be used to identify Arabic NEs without the need for linguistic analysis.

7.2 List Lookup Features

These features are used to classify the identity of the target word with respect to its membership in various lists, called word-identity features by Farber et al. (2008). In Table 5, we present four important categories of lists used in the literature as binary discriminative features indicating whether a word is a member of any of these lists. Gazetteer list inclusion is a direct way to express a typical NE.

The Lexical Trigger list provides a way to identify entity cues or predictive words, such as the relation between a person and a title (e.g., الحاسوبية عماد زيتوني أستاذ اللسانيات, *Professor of Computational Linguistics Imed Zitouni*), whereas the Blacklist

Table 5
List lookup features.

Feature	Description
Gazetteer	A binary feature indicating the existence of the word in an individual gazetteer.
Lexical Trigger	A binary feature indicating the existence of the word in the individual lexical trigger list.
Blacklist	A binary feature indicating the non-existence of the word in an individual blacklist.
Nationality	A binary feature indicating the existence of the word in the nationality list.

(e.g., أستاذ اللسانيات الحاسوبية رئيس المؤتمر, *Professor of Computational Linguistics chairman of the conference*) counterindicates the presence of an NE as a means of resolving the ambiguity of words in the ambiguous position.

Many authors have proposed a way to recognize nationality by identifying relevant word forms that are frequently used in NEs and their context, e.g., الجامعة الأردنية (*The Jordanian University*) and رانيا الملكة الأردنية (*the Jordanian queen Rania*), respectively. Nationality word forms can be stemmed to a country name using a country gazetteer and well-known affixes in the rule-based approach (Shaalan and Raza 2008), for example, الجامعة الأردن[ية] (*Jordan[ian] University*); or they may be searched using a separate closed list in the ML approach (Benajiba, Diab, and Rosso 2008b), for example, *Jordanian* in this list might be expressed by the forms أردني, أردنية, الأردني, or الأردنية.

7.3 Contextual Features

Contextual features are local features defined over the targeted word and include the type of words that occur with the NEs, namely, left and right neighbors of the candidate word which carry effective information for the identification of NEs. Table 6 lists subcategories of these features. Usually, they are defined in terms of a sliding window of tokens/words. For example, if the size of the sliding window is 5, the decision on the targeted word is made based on its features as well as the features of its two immediate left and right neighbors (i.e., +/- 2 words Abdallah, Shaalan, and Shoaib 2012). Different window sizes have been used with contextual features. For example, in Benajiba, Diab, and Rosso (2008b) the window size was +/- 1, whereas in Benajiba et al. (2010) it was +/- 1 to 3. The sliding step over the text, which refers to the interval between two adjacent sliding windows, should also be defined: usually it is 1. In the literature, contextual features specifically describe word *n*-gram and rule-based features.

Word *n*-gram contextual features can be derived from the context of a document in order to extract the relationships between previously identified NEs and an encountered word within the input document (Benajiba, Diab, and Rosso 2008b). They are used to investigate the space of the surrounding context for the NEs by taking into account the features of a window of words surrounding a candidate word in the recognition process.

Rule-based features are contextual features that are derived from rule-based decisions. Abdallah, Shaalan, and Shoaib (2012) suggested that these features have a critical impact on the performance of pure ML-based NER components in particular, and proposed hybrid systems combining rule-based with ML-based components in general. In this system, an *n*-word sliding window is used for each word in corpus. Table 7 provides sample instances of these features for a window of size 5.

Table 6
Contextual features.

Feature	Description
Word *n*-gram	The features of a sliding window comprising a word *n*-gram that includes the candidate word, along with preceding and succeeding words.
Rule-based	The features of a sliding window derived from rule-based NER decisions.

Table 7
Sample rule-based features for 5-word window.

Targeted Word	English Tran.	Wi-2	Wi-1	Wi	Wi+1	Wi+2
الرئيس	*President*	OTHER	OTHER	OTHER	OTHER	Person
الروسي	*Russian*	OTHER	OTHER	OTHER	Person	Person
فلاديمير	*Vladimir*	OTHER	OTHER	Person	Person	OTHER
بوتين	*Putin*	OTHER	Person	Person	OTHER	OTHER

7.4 Language-Specific Features

These features are related to certain aspects of the Arabic language. Table 8 lists subcategories of language-specific features. They specifically describe part-of-speech (POS), morphological features, and base-phrase chunks (BPC).

Arabic words generally carry rich morphological information (Marton, Habash, and Rambow 2010), some of which includes noun–adjective agreement and special markings indicating nominals in compounds. The MADA toolkit has been found to be very useful in generating a number of informative language-specific features for each input word (Habash, Rambow, and Roth 2009). One of these features is the POS morpho-syntactic tag, which plays a significant role in Arabic NLP. An Arabic NE usually consists of either noun (NN) or proper noun (NNP) tags. In Benajiba and Rosso (2007), very good results were obtained using the POS tagging feature, which was exploited to improve NE boundary detection. The shared task of CoNLL now includes a POS column in its corpora. Thus, the POS tag is a good distinguishing feature for Arabic NEs; it has been studied separately in the literature to determine its impact on NER. As an example, Farber et al. (2008) demonstrated a significant improvement in Arabic NER using a POS feature. In order to make use of the varying importance of different morphological features, a careful choice of relevant features and their associated value representations have to be taken into consideration when studying Arabic NER. Benajiba, Diab, and Rosso (2008b) report on the impact of morphological features that affect NEs, such as aspect, person, definiteness, gender, and number.

The structure of an Arabic sentence allows different arrangements of NEs: NEs may appear anywhere in the sentence and at different distances from lexical triggers. Elsebai, Meziane, and Belkredim (2009) and Elsebai and Meziane (2011) point out that these arrangements might complicate the structure of the induced heuristics rules of their rule-based NER system. This observation has led to using the BPC feature as an indicator of embedded NEs (Benajiba and Rosso 2008). BPC features are related to the type of words that occur with NEs and their syntactic relations (Benajiba, Diab, and Rosso 2008b). They are usually identified by shallow syntactic parsing. The Amira toolkit has been found to be very useful in generating BPC features (Diab 2009).

7.5 Feature Selection

It is useful to think of the ML-based NER as consisting of four major steps: 1) *feature selection*; 2) *algorithm selection* or the decision of which ML algorithm(s) to use for

Table 8
Language-specific features.

Feature	Description
POS	The label identifying the part-of-speech category of a word.
Morphological	A set of morphological information (excluding POS).
BPC	Phrase-level labels identifying syntactic chunks such as noun phrases (NPs) and verb phrases (VPs) within a text.

training and classification; 3) *training*, the actual learning of distinguishing patterns using the selected feature list; and 4) *classification*, applying these patterns to the input text to detect and classify the NEs.

The success of a learning algorithm is crucially dependent on the features it uses. A supervised learning algorithm uses an annotated corpus. The training set derived from an annotated corpus represents the NEs in terms of feature values.

Feature selection refers to the task of identifying a useful subset of features chosen to represent elements of a larger set (i.e., the feature space). The selection of the subset to be utilized by a classifier is a very critical issue and when optimized it can enhance the performance of a system dramatically (Nadeau and Sekine 2007). The main purpose of this step is to try to find a strong correlation between an NE and one or more combined features in order to explore generalizations over the set of selected features. Iterative experiments are conducted to gain a better understanding of different combinations of the selected features and their impact on the NER task. In a typical learning environment, reporting experiments with all the different combinations of features would adversely affect the readability of the achieved results (Abdul-Hamid and Darwish 2010). So, in the literature, the presentation highlights experiments that their enabled feature combination show significant (or best) obtained results for the evaluation data sets.

Under each type of feature, there is a set of characteristics that need to be considered and the methods used to extract them may differ in their degree of accuracy. If all feature values and their combinations are selected the feature space becomes high-dimensional. Not all features are equally important for the recognition task. Thus, even the set of selected features needs to be evaluated in order to find the optimal feature set for an NER system. There are different ways to carry out feature selection.

The most widely used method is to select features manually by a process of enabling features one by one to determine their effects. Another method is to initially decide on the feature set by testing features in isolation at the beginning, and incrementally combining them in different sets until a set containing all the features is reached and is tested. Benajiba, Diab, and Rosso (2008a) and Benajiba, Diab, and Rosso (2008b) used an incremental approach that selects the top *n* features. Then, the features are ranked in a decreasing order according to their individual impact (using the F-measure obtained for each NE), keeping only the set that yields the best results at each iteration.

8. Tools for Developing Arabic NER Systems

A good number of tools are available for developing and evaluating Arabic NER systems, allowing for easy replicability of experiments. The following is a non-exhaustive list of NER tools that have been used in the Arabic NER literature. The tools can be classified into three categories according to their functions: Integrated Development Environments tools, ML tools, and Arabic NLP tools.

8.1 Integrated Development Environments

GATE[12] (The General Architecture for Text Engineering): This is one of the most popular freely available software tools dealing with NLP. GATE is a suite of Java tools that provides an infrastructure for developing and deploying software components that process human language (Maynard et al. 2000; Cunningham 2002; Cunningham et al. 2011). The motivating factors behind the development of GATE include reusability of components, task-based evaluation, comparative evaluation, collaborative research, robustness, efficiency, and portability; the tools support nine languages (English, French, German, Italian, Chinese, Arabic, Romanian, Hindi, and Cebuano). GATE provides a set of essential tools for NLP system development, including tokenizers, gazetteers, POS taggers, chunkers, and parsers. It facilitates the development of rule-based NER systems by providing the user with the capability of implementing grammatical rules as a finite state transducer using JAPE. It also has an Arabic plug-in that contains a tokenizer, gazetteers, an OrthoMatcher component, and a grammar, all of which are used within a simple Arabic rule-based NER application built as a part of GATE. GATE can be used to extract basic entities, such as date, name, location, organization, and so on. A number of scholars have used the GATE environment in their research studies on Arabic NER, including Maynard et al. (2002), Elsebai, Meziane, and Belkredim (2009), Elsebai and Meziane (2011), and Abdallah, Shaalan, and Shoaib (2012).

NooJ:[13] This is a freely available linguistic development environment for many languages. NooJ allows the developer to construct, test, and maintain large coverage lexical resources, as well as apply morpho-syntactic tools for Arabic processing. It can recognize all Unicode encodings, which is a very important feature for processing Arabic Script languages. NooJ can recognize rules written in finite-state form or context-free grammar form, facilitating the development of rule-based NER systems. Nooj provides a disambiguation technique based on grammars to resolve duplicate annotations. Arabic is one of the languages that are supported by NooJ; there are free Arabic resources for use within the NooJ environment on the NooJ official Web site. Mesfar (2007) has used NooJ in his Arabic NER research.

LingPipe:[14] A toolkit for text engineering and processing, the free version has limited production capabilities and one must upgrade in order to obtain full production abilities. The toolkit is language-, domain-, and genre-independent. It supports the development of different language processing tasks such as POS tagging, spelling correction, NE recognition, and word sense disambiguation. The NER component is based on hidden Markov models and the learned model can be evaluated using *k*-fold cross validation over annotated data sets. LingPipe recognizes corpora annotated using the IOB scheme. The LingPipe NER system has been applied by ANERcorp to demonstrate how to generate a statistical NER model for Arabic; the

[12] GATE is available at `http://gate.ac.uk/`.
[13] NooJ is available at `http://www.nooj4nlp.net`.
[14] LingPipe is available at `http://alias-i.com/lingpipe/`.

details and results are presented on the toolkit's official Web site. AbdelRahman et al. (2010) used ANERcorp to compare their proposed Arabic NER system with LingPipe's built-in NER.

8.2 Machine Learning Tools

In the Arabic NER literature, the ML tools of choice are data-mining-based tools that support one or more ML algorithms, such as Support Vector Machines (SVM), Conditional Random Fields (CRF), Maximum Entropy (ME), hidden Markov models, and Decision Trees. These tools are YASMET, CRF++, YamCha, and WEKA. They all share the following features: a generic toolkit, language independence, absence of embedded linguistic resources, a requirement to be trained on a tagged corpus, the performance of sequence labeling classification using discriminative features, and a suitability for the pre-processing steps of NLP tasks.

YASMET:[15] This free toolkit, which is written in C++, is applicable to ME models. The toolkit can estimate the parameters and computes the weights of an ME model. YASMET is designed to handle a large set of features efficiently. However, there are not many details available about the features of this toolkit. In Benajiba, Rosso, and Benedí Ruiz (2007), Benajiba and Rosso (2007), and Benajiba, Diab, and Rosso (2009a), YASMET was used to implement ME approach in Arabic NER.

CRF++:[16] This is a free open source toolkit, written in C++, for learning CRF models in order to segment and annotate sequences of data. The toolkit is efficient in training and testing and can produce *n*-best outputs. It can be utilized in developing many NLP components for tasks such as text chunking and NER, and can handle large feature sets. Both Benajiba and Rosso (2008), Benajiba, Diab, and Rosso (2008a, 2009a), and Abdul-Hamid and Darwish (2010) have utilized CRF++ to develop CRF-based Arabic NER.

YamCha:[17] A commonly used free open source toolkit written in C++ for learning SVM models. This toolkit is generic, customizable, efficient, and has an open source text chunker. It has been utilized to develop NLP pre-processing tasks such as NER, POS tagging, base-NP chunking, text chunking, and partial chunking. YamCha performs well as a chunker and is capable of handling large sets of features. Moreover, it allows for redefining feature parameters (window-size) and parsing-direction (forward/backward), and applies algorithms to multi-class problems (pair wise/one vs. rest). Benajiba, Diab, and Rosso (2008a), Benajiba, Diab, and Rosso (2008b), Benajiba, Diab, and Rosso (2009a), and Benajiba, Diab, and Rosso (2009b) have used YamCha to train and test SVM models for Arabic NER.

Weka:[18] A collection of ML algorithms developed for data mining tasks. The algorithms can either be applied directly to a data set or called from your own Java code. The toolkit contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It has also been found useful for developing new ML schemes (Witten, Frank, and Hall 2011). The Weka workbench supports the use of *k*-fold cross validation with each classifier and the presentation of results by means of standard Information Extraction measures. Most recently,

[15] http://www-i6.informatik.rwth-aachen.de/web/Software/YASMET.html.

[16] http://crfpp.sourceforge.net/.

[17] http://chasen.org/∼taku/software/yamcha/.

[18] http://www.cs.waikato.ac.nz/ml/weka/.

Abdallah, Shaalan, and Shoaib (2012) and Oudah and Shaalan (2012) have successfully used Weka to develop an ML-based NER classifier as part of a hybrid Arabic NER system.

8.3 Arabic NLP Tools

The complexity of Arabic morphology makes it a very challenging research topic. In this section we present Arabic morpho-syntactic pre-processing tools that are widespread and used extensively in the Arabic NER literature, including BAMA, MADA, and the AMIRA toolkit.

BAMA (Buckwalter Arabic Morphological Analyzer).[19] BAMA is one of the most widely used Arabic NLP tools and is widely cited in the literature (Buckwalter 2002; Elsebai and Meziane 2011). It contains over 80,000 words, 38,600 lemmas, three dictionaries (Prefix, Stem, Suffix), and three compatibility tables (Prefix-Stem, Stem-Suffix, Prefix-Suffix) (Habash 2010). Entries of the stem dictionary include English glosses, which have been used to disambiguate NEs. BAMA output lends itself to information extraction and retrieval processing as it takes an input Arabic word and returns a stem rather than a root. The word is selected with or without short vowels. Then it is segmented and compatibility-checked for the correct combination of its segments, producing all possible analyses of the input word. BAMA transliteration of the output makes it readable; this is more useful for readers who do not have the ability to read the Arabic script but are familiar with Latin script. In addition, the transliteration[20] output can be converted directly to Unicode Arabic with a minimal amount of automatic processing. BAMA has been made available through the Linguistic Data Consortium. Some of the Arabic NER studies that rely on BAMA for performing morphological analysis include Farber et al. (2008), Elsebai, Meziane, and Belkredim (2009), and Al-Jumaily et al. (2012).

(MADA+TOKAN).[21] MADA stands for Morphological Analysis and Disambiguation for Arabic. The combined package is built on top of BAMA as a natural successor that builds on prior successes and meets the growing requirements of many Arabic NLP applications (Habash, Rambow, and Roth 2009). The package consists of two components. Morphological analysis and disambiguation are handled in the MADA component. Morphological analysis also supports the ability to tokenize and stem deterministically. Because there are many different ways to tokenize Arabic (tokenization is a convention adopted by researchers), the TOKAN component allows the user to specify any tokenization scheme that can be generated from disambiguated analyses. The MADA+TOKAN package provides one solution to all of the basic problems in Arabic NLP, including tokenization (the segmentation of clitics from a word with attendant spelling modifications), diacritization (insertion of disambiguating short-vowel diacritics), morphological disambiguation (determining the full morphological information for each word given its context), POS tagging (determining specific morphological information for each word), stemming (reducing each word to its base form), and lemmatization (determining the citation form lemma of the set of word lexemes

[19] LDC Catalog No.: LDC2004L02, on http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2004L02.

[20] See the BAMA mapping table at http://www.qamus.org/transliteration.htm.

[21] MADA+TOKAN constitute a single package that is continuously updated. The system is freely available for research purposes at http://www1.ccls.columbia.edu/MADA/MADA_download.html.

to which each word in the data belongs). MADA operates by examining a list of all possible analyses for each word generated by BAMA, and then selecting the analysis that best matches the immediate context by means of SVM models. This classifier uses 19 distinct and weighted morphological features to provide complete diacritic, lexemic, glossary, and morphological information (Habash 2010). However, because MADA is built on top of BAMA, it inherits all of BAMA's limitations. For example, if no analysis is given by BAMA, no lemmatization or diacritization is undertaken. It has been noted in the literature that because MADA was trained and tested on the Penn Arabic Treebank (Maamouri et al. 2004), its coverage and quality relative to other text types has not yet been evaluated (Attia et al. 2010; Mohit et al. 2012). The richness of MADA's extracted morphological features has been exploited by Arabic NER studies such as those carried out by Farber et al. (2008), Benajiba and Rosso (2008), Benajiba, Diab, and Rosso (2008a), Benajiba, Diab, and Rosso (2009a), Benajiba, Diab, and Rosso (2009b), Oudah and Shaalan (2012), and Oudah and Shaalan (2013).

AMIRA.[22] A statistical Arabic processing toolkit that includes a clitic tokenizer, POS tagger, and BPC or shallow syntactic parser (Diab 2009). It has been widely used for different NLP applications due to its speed and high performance. BPC is one of the distinctive characteristics of this toolkit. AMIRA has been used in the extensive studies of Arabic NER by Benajiba, Diab, and Rosso (2008a), Benajiba, Diab, and Rosso (2008b), Benajiba, Diab, and Rosso (2009a), and Benajiba, Diab, and Rosso (2009b).

9. Evaluation

The main objective of evaluation is to rank NER systems based on the ability to annotate a text in the way that an Arabic linguist would. For any research undertaking, it is necessary to evaluate the system's results with respect to existing systems on the assumption that the same reported results should be replicated under the same experimental settings (Kumaran, Khapra, and Li 2010). Results are easily compared when they utilize the same standard evaluation corpora, where every NE has a type assigned to it.

CoNLL's evaluation metrics are used in the Arabic NER literature. These are aggressive metrics that do not assign partial credit: An exact match of the NE as a whole and a correct classification must be identified in order to earn credit. The reason that this method of scoring is popular is due to its simplicity in calculating and analyzing results. NER systems are compared based on the standard micro-averaged F-measure with the Precision being the ratio of the detected NEs that are correctly classified by the system, and the Recall being the ratio of the relevant NEs that are detected by the system (Yang 1999). Mesfar (2007) has redefined the evaluation measures to account for partially correct NE tagging that arises due to a lack of information about unknown words within NEs. No other research has accepted this additional parameter of the evaluation measures.

High Recall means that the system returned most of the relevant results, whereas high Precision means that the system returned more relevant results than irrelevant. Often, there is an inverse relationship between Precision and Recall, where it is possible to increase one at the cost of lowering the other. Recently, Mohit et al. (2012)'s exploration of the Recall–Precision tradeoff proposed a Recall-oriented learning method that

[22] A demo of the system is available at `http://nlp.ldeo.columbia.edu/amira/`.

improved Recall over Precision during semi-supervised discriminative learning of NEs from Wikipedia.

K-fold cross validation is usually adopted with the scoring method in order to avoid over-fitting. The data set is randomly divided into *k* folds of equal size. Each fold is used as a testing set and the remaining folds are used as a training set, and then the test results (i.e., F-measure, Precision, Recall) are averaged over the rounds. When comparing evaluation results it is important to replicate the same split for training and testing because different splits can have significant effects on the Precision and Recall values (Benajiba et al. 2010). Characteristics of splits include the size of training and test data sets, ratio of NEs, number of NEs, and average length of NEs (Benajiba, Diab, and Rosso 2008a). The advantage of the cross-validation method over other methods, such as repeated random sub-sampling or the percentage split method (holdout), is that all observations are used equally for both training and validation, and each observation is used for validation exactly once. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. Typically, *10-fold* cross-validation is used, but in general *k* remains a variable parameter.

10. NER Systems

The importance of Arabic NER systems has been well recognized by the community, as evidenced by the noteworthy publications in this important area. In this section we present different NER systems. They are classified according to the approach used. Unfortunately for the research community, most of the efforts to develop reliable Arabic NER systems have been undertaken for commercial purposes (Benajiba, Rosso, and Benedí Ruiz 2007; Zaghouani 2012). Because information on the specifications and performance of these systems is generally not available, it is difficult to carry out a fair comparison of the performance of these systems relative to the systems proposed by the Arabic NER research community. Examples of commercial Arabic NER systems are: ANEE[23] (Coltec), IdentiFinder[24] (BBN), NetOwlExtractor[25] (NetOwl), Siraj[26] (Sakhr), Clear Tags[27] (ClearForest), Enterprise Search[28] (FAST ESP), and InXight-Smart-Discovery-Entity-Extractor[29] (InXight).

10.1 Rule-Based Systems

Rule-based NER systems depend mainly on hand-made linguistic rules (i.e., grammars) defined by linguists. In the literature, the development of systems using the rule-based approach was motivated mainly by the fact that the architecture of the available NER development tools was optimized for building rule-based systems. The approach compensates for the lack of Arabic NER linguistics resources, and is favored based on the encouraging results obtained by various Arabic rule-based systems as shown in this section. Experiments for reporting the performance of rule-based systems are described

[23] http://www.coltec.net/Portals/0/COLTEC_PDFs/ANEE_NEW.pdf.
[24] http://www.bbn.com/technology/speech/identifinder.
[25] http://www.sra.com/netowl/entity-extraction/.
[26] Online demo version available at http://siraj.sakhr.com/.
[27] http://www.clearforest.com/solutions.html.
[28] http://www.microsoft.com/enterprisesearch.
[29] http://www.inxightfedsys.com/products/sdks/tf/default.asp.

at three levels: the NE type, the level of linguistic knowledge (morphology and syntax), and the inclusion/exclusion of gazetteers. A corpus is often needed to evaluate an NER system, but not necessarily for its development. This is the reason that many of these experiments are based on a non-standard data set that has been acquired by developers for evaluation purposes.

Maloney and Niv (1998) presented the TAGARAB system, an early attempt to handle Arabic rule-based NER. The system identifies the following NE types: person, organization, location, number, and time. A morphological analyzer is used to decide where a name ends and the non-name context begins. For evaluation, 14 texts from the AI-Hayat CD-ROM were selected randomly and manually tagged. The overall performance obtained 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 NER system that makes use of lexical triggers. Some special verbs, such as أعلن (*announce*), is used to predict the positions of names in the Arabic sentence. The research assumes that an NE appears close to lexical triggers no more than three words from the cue word and that the NE has a maximum length of seven words. Some names may be attached to different types of lexical triggers and to more than one lexical trigger in the same phrase. For example, the phrase الدكتور خالد شعلان رئيس قسم تكنولوجيا المعلومات (*Dr. Khaled Shaalan the Chairman of IT Department*) has the lexical triggers الدكتور (*Dr*) and رئيس قسم (*Chairman Department*). In Abuleil's (2004) work, Arabic NER is part of a question-answering system. The system starts by marking the phrases that could include names. Afterwards, it builds up a graph that represents the words in these phrases and the relationships between them. Finally, rules are applied to classify and generate the NEs before saving them in a database. The system has been evaluated on 500 articles from the *Al-Raya* newspaper, published in Qatar. It obtained a Precision of 90.4% on persons, 93% on locations, and 92.3% on organizations.

Samy, Moreno, and Guirao (2005) used comparable corpora in Spanish and Arabic and an NE tagger. A mapping technique is used to transliterate words in the Arabic text and return those matching with NEs in the Spanish text as NEs in Arabic. The Spanish NE tags are used as indicators for tagging the corresponding NEs in the Arabic corpus. Exceptions arise when it tries to recognize NEs whose Arabic equivalents are completely different, such as *Grecia* (*Greece*) اليونان, or do not have a precise transliteration, such as *Somalia* الصومال. An experiment was conducted using 1,200 sentence pairs. In another experiment, a stop word filter was additionally applied to exclude the stop words from the potential transliterated candidates. The filter improved the overall Precision from 84% to 90%; the Recall was very high at 97.5%.

Mesfar (2007) used NooJ to develop a rule-based Arabic NER system. The system identifies the following NE types: person, location, organization, currency, and temporal expressions. The Arabic NER is a pipeline process that goes through three sequential modules: a tokenizer, a morphological analyzer, and Arabic NER. Morphological information is used by the system to extract unclassified proper nouns and thereby enhance the overall performance of the system. An evaluation corpus was built from Arabic news articles extracted from the *Le Monde Diplomatique* newspaper. The reported results based on individual NE types were as follows: Precision, Recall, and F-measure range from 82%, 71%, and 76% for Place names to 97%, 95%, and 96% for Time and Numerical expressions, respectively.

Another system adopting the rule-based approach for identifying person names is PERA (Shaalan and Raza 2007). This research describes the structure of Arabic personal names: 'ism', 'kunya', 'nasab', 'laqab', and 'nisba'. The 'ism' is a proper name given shortly after birth (i.e., the given name). Examples of such names are محمد (Muhammad, *Mohammed*), موسى (Musa, *Moses*), إبراهيم (Ibrahim, *Abraham*).

The 'kunya' is an honorific name or surname that states the name of someone's father (أبو, Abu) or mother (أم, Umm). For example: أبو داود (Abu Da'ud, *the father of David*), أم سليم (umm Salim, *the mother of Salim*). When using a person's full name, the 'kunya' precedes the given name, for example, أبو يوسف حسن (Abu Yusuf Hassan, *the father of Joseph, Hassan*), أم جعفر أمينة (Umm Ja'far Aminah, *the mother of Ja'far, Aminah*).

The 'nasab' indicates the person's heritage by the word إبن Ibn (colloquially and MSA, بن Bin), which means *son* (بنت Bint for *daughter*): for example, إبن عمر (Ibn 'Umar, *the son of Omar*), بنت عباس (bint 'Abbas, *the daughter of Abbas*). The 'nasab' follows the 'ism' in usage, for example, حسن إبن فراج (Hasan Ibn Faraj, *Hasan the son of Faraj*), سمية بنت خبيت (Sumayya Bint Khubbat, *Sumayya the daughter of Khubbat*). Many historical persons are more familiar to us by their 'nasab' than by their 'ism'. Notable examples are: the historian ابن خلدون (*Ibn Khaldun*), the traveler إبن بطوطة (*Ibn Battuta*), and the philosopher إبن سينا (Ibn Sina, *Avicenna*).

A 'laqab' is a combination of words into a byname or epithet, usually religious or relating to a trait, a descriptive, or some admirable quality the person had or would like to have. Examples are: الرشيد (Al-Rashid, *the Rightly guided*), and الفاضل (Al-Fadl, *the Prominent*). In practice, 'laqabs' follow the 'ism', for example, هارون الرشيد (Harun Al-Rashid, *Aaron the Rightly guided*).

Finally, a 'nisba' is a name derived from a person's: trade or profession, place of residence or birth, or religious affiliation. Examples are: الحلاج (Al-Hallaj, *the dresser of cotton*), المصري (Al Msri, *The Egyptian*), إسلامي (Islami, *Islamic*). Nisbas follow the 'ism' or, if the name contains a 'nasab' (of however many generations), generally follow the 'nasab.'

In PERA, rules use regular expressions that include these naming constituents to recognize person names, where "+" indicates one or more elements; "\s" represents white space; "|" represents alternatives; and "?" represents an optional element. For example, consider the following rule:

$$
\begin{array}{l}
((\mathit{honorific\_trigger}^{+}\backslash s((\text{ال})?\mathit{location\_GAZ}(\text{ي|ية})^{+}\backslash s)?)^{+}\\
\mathit{first\_name\_GAZ}(\backslash s^{+}\mathit{last\_name\_GAZ})?\backslash s^{+}(\mathit{number})?\\
)
\end{array}
$$

This rule recognizes a person name such as الملك الأردني عبد الله الثاني (*The Jordanian king Abdullah II*) that is composed of a first name followed by optional last name, which in turn is followed by an optional ordinal number based on preceding person triggers. The triggers are the honorific الملك (*the king*) and the 'nisba' الأردني (*Jordanian*). 'Nisba' is represented by the expression (ال)? location_GAZ (ي|ية) that indicates a nationality (masculine or feminine) adjective as in [الاردن][يـ][ية] (*Jordan[ian]*) and [الـ][مصر][يـ][ـة] (*[The] Egypt[ian]*).

The system consists of three components: gazetteers, grammar rules, and a filtering mechanism. Whitelists of person names are provided in the gazetteers component in order to extract the exact matching of NEs regardless of the grammar. Afterwards, the input text is presented to the grammar in order to identify other person NEs. Finally, the filtering mechanism is applied to NEs in order to exclude invalid person names. PERA was evaluated using the ACE and Treebank Arabic data sets and obtained 85.5%, 89%, and 87.5% for Precision, Recall, and F-measure, respectively.

As a continuation of the research carried out by Shaalan and Raza (2007), a NERA system was introduced by Shaalan and Raza (2008, 2009) that generalizes the findings from PERA. NERA addresses major challenges posed by NER in the Arabic language arising from the complexity of the morphological system, peculiarities in the Arabic orthographic system, non-standardization of the written text, ambiguity, and lack of resources. The system identifies the following NE types: person, location, organization, date, time, ISBN, price, measurement, phone numbers, and filenames. NERA used the FAST ESP[30] framework, whose architecture is optimized for rule-based systems, as an implementation platform. Like PERA, the NERA system has three components (gazetteers, local grammars in the form of regular expressions, and a filtering mechanism). Gazetteer entries include English transliterations, an important feature for cross-lingual and multilingual applications. The evaluation is based on manually constructed corpora from ACE, the Web, and organizations. NERA obtained an F-measure of 87.7% for person, 85.9% for locations, and 83.15% for organizations.

Traboulsi (2009) presented a ruled-based approach for person NER that uses a local grammar and dictionaries. The extraction process is based on reporting verbs that can be used within grammars to indicate one or more NE types. In the following example, two NEs (person and organization names) can be recognized using verb and job title triggers.

... ان [اوبك] ORG_Name [رئيس] TITLE_trigger [أحمد الفهد الصباح] Person Name قال ...
... said [Ahmad Al-Fahd Al-Sabah] [OPEC]'s [president] that ...

Notice that not all verbs that occur before person names can correctly identify NEs. For example, in the following sentence إتهم صدام بوش (*Saddum accused Bush*, accused Saddum Bush), using the verb as a trigger would result in the extraction of صدام بوش (*Saddum Bush*) as a name although these are in fact two different names, corresponding to the subject and object of the verb, respectively. An analytical study was conducted by Traboulsi (2009) for his own corpus (arabiCorpus) that was collected from several newspapers, books, the Quran, and some medieval medical and philosophical texts. The study addressed frequency, collocation, and concordance analyses of the corpus. No substantive evaluation results were reported.

Elsebai, Meziane, and Belkredim (2009) and Elsebai and Meziane (2011) have proposed a rule-based person name recognition system. The system is implemented using GATE. Heuristic rules make use of two kinds of lexical triggers in the Arabic text. An introductory verb trigger, for example, قال (*said*), identifies the phrases that probably include person names. An NE trigger, for example, طبيب (*doctor*), a job title, identifies a person name within phrases. The structure of the heuristic rule depends on the relative position of each kind of lexical trigger in the input text and its position relative to other words. BAMA (Buckwalter 2002) has been integrated to extract the morphological features of the target word that are used within rules to identify whether the target word is a proper noun. This has led to the elimination of the need for any predefined person name gazetteers. Name lists, specifically, place and organization names, and stop words, such as prepositions, which occur after lexical triggers, are used to counter-indicate the presence of a person name. For example, although أبو ظبي (*Abu Dhabi*) in the phrase أعلنت أبو ظبي عن الفائزين (*Abu Dhabi announced the winners*) is recognized as a proper noun, it is discarded because it belongs to the list of places and hence should not be

[30] FAST ESP is a product of the FAST Search & Transfer Company, which was acquired by Microsoft in 2008.

recognized as a person name. Two experiments were conducted (Elsebai, Meziane, and Belkredim 2009; Elsebai and Meziane 2011). The first experiment used around 700 news articles extracted from an Arabic media Web site, and the second used 500 articles. The overall system performance in the first experiment was 93%, 86%, and 89%, for Precision, Recall, and F-measure, respectively; the overall performance in the second experiment was 88%, 90%, and 89%, for Precision, Recall, and F-measure, respectively.

Alkharashi (2009) described the formation of an Arabic person name from root and pattern using the traditional Arabic morphology and suggested relevant computational resources. The author introduced a set of database tables in order to assist Arabic NER: root-pattern, a frequency list of roots, and lexical trigger tables. A corpus was created from Saudi person names with specific person name tags: root of person NE, features indicating the possibility of affixation, and gender characteristics. The main objective was to recognize the constituents of the person NE, these being the simple form, the affix, and connectors. For example, the name of the Umayyad caliphate الوليد بن عبد الملك (*Al-Waleed bin Abd Al-Malik*) has ملك (*Malik*) and وليد (*Waleed*) as simple names, عبد (*Abd*) and ال (*Al*) as name prefixes, and بن (*Bin*) as a name connector. The study has reported interesting observations about features of highly frequent patterns and their lengths. A simple test for assessing how well the pattern of a person name was recognized was conducted on 60,000 generated person names entries. It demonstrated that the correct pattern appears 94% of the time as one of the first three suggested patterns, 86% as one of the first two suggested patterns, and 69% of the time as the first suggested pattern.

Al-Shalabi et al. (2009) presented an Arabic NER algorithm for retrieving Arabic proper nouns using lexical triggers. The research takes into consideration regional patterns such as the name connector ولد (ould, *son of*) used in Mauritanian person names (e.g., مختار ولد داداه, *Moktar Ould Daddah*). The algorithm identifies the following NE types: people, major cities, locations, countries, organizations, political parties, and terrorist groups. However, the reported research only focuses on person NEs. The algorithm uses heuristic rules to preprocess the input to clean the data and remove affixes. Then, internal evidence triggers, such as person name connectors, are used to recognize the NEs. The system was evaluated using 20 randomly selected documents from the *Al-Raya* newspaper published in Qatar, and the *Alrai* newspaper published in Jordan. An overall precision of 86.1% was observed.

Attia et al. (2010) proposed a method for acquiring a richer NE lexicon using Arabic WordNet (Elkateb et al. 2006) and Arabic Wikipedia.[31] The proposed NE lexicon enhances the lexical entries in WordNet and produces a well-structured Arabic NE lexical resource. The main objective is to extract Arabic WordNet's instantiable nouns and to identify the corresponding categories in the Arabic Wikipedia. These categories act as lexical triggers. A decision is made in order to identify which of the Wikipedia articles of these categories correspond to NEs. They are then extracted, connected to Arabic WordNet, and inserted in the NE repository. In a subsequent post-processing step, further NEs are acquired by exploiting inter-lingual links. Finally, the NEs acquired are diacritized. This lexical resource is useful for Arabic NER; the results are not only recognized (tagged) NEs but also identified synsets which are semantically related to them (synonyms, subtypes, supertypes, etc.). Likewise, Abouenour, Bouzoubaa,

[31] WordNet (Fellbaum 2005) is a large lexical database that originally implemented for English. Arabic WordNet is still a limited lexical resource. On the other hand, Wikipedia is a popular ubiquitous source of corpus data for information extraction because of its size, currency, rich semi-structured content, diverse topics, and closer resemblance to web text than newswire (Balasuriya et al. 2009; Mohit et al. 2012).

and Rosso (2010) suggested enriching the NEs in Arabic WordNet by using an ontology-based method. This is used in the query expansion stage of an Arabic Question Answering system (Lahsen, Bouzoubaa, and Rosso 2012), resulting in an improvement of the ranking of the returned passages.

Shihadeh and Neumann (2012) proposed an Arabic NER system called ARNE, which recognizes person, location, and organization NEs based only on a gazetteer lookup approach; the system provides morphological information using a system called ElixirFM, developed by Smrz (2007). The system does not use any rules or context information for Arabic NER. Before recognizing the NEs, ARNE carries out three pre-processing steps that are not used by the gazetteer lookup approach: tokenization, Buckwalter transliteration, and POS tagging. ARNE uses the ANERgazet gazetteer that was developed by Benajiba, Rosso, and Benedí Ruiz (2007) and Benajiba and Rosso (2007). ARNE can recognize a NE that has a maximum length of four words. The experimental results obtained low overall performance: 38%, 27%, and 30% for Precision, Recall, and F-measure, respectively. The authors suggest several reasons as to why the F-measure did not achieve higher values. These include the size and quality of the gazetteers, the richness and complexity of Arabic morphology, and the ambiguity problem inherent in Arabic NEs.

Al-Jumaily et al. (2012) proposed a rule-based NER system that can be used in Web applications. The system identifies the following NE types: person, location, and organization NEs. The system was developed using GATE and provides Arabic morphological analysis in a method similar to BAMA. It also integrates different gazetteers from GATE, DBPedia,[32] and ANERGazet.[33] The system was evaluated using ANERcorp. Two experiments were carried out to study the effect of Arabic prefixes and suffixes on the recognition results. If an Arabic token (prefix-stem-suffix) is recognized, then a verification process is used to ensure the compatibility between the three possible combinations (prefix-stem, stem-suffix, and prefix-suffix). The verification process has improved the recognition results of NEs across all types, although these improvements were not symmetrical. The improvements in the Precision of person, location, and organization are 7.32%, 5.55%, and 5.14%, respectively. Suggestions for improvements include: 1) adding new patterns to the system's dictionary, 2) accounting for all transliteration variants of Latin names, 3) adopting semi-automatic methods to tag unrecognized words, and 4) performing contextual analysis to resolve ambiguity arising from words that may belong to different entity types (e.g., whether باريس (*Paris*) is a location or person).

Zaghouani et al. (2010) presented an adaptation of a multilingual system, the Europe Media Monitor (EMM) Information Retrieval and Extraction application NewsExplorer[34] (Steinberger, Pouliquen, and Van der Goot 2009), to consider Arabic. This system at present includes 19 languages and is able to analyze large volumes of news text. The EMM-NewsExplorer architecture is optimized for ruled-based systems. The adaptation resulted in a rule-based Arabic NER system (RENAR; Zaghouani 2012), which uses a handwritten set of language-independent rules (Steinberger, Pouliquen, and Ignat 2008) in combination with specific resources for Arabic. Rules are described using the following notations: "\w+" for an unknown word, "\b"

[32] See `http://dbpedia.org/About`. Entries of the DBPedia are translated from English to Arabic using Google Translate.

[33] See `http://users.dsic.upv.es/grupos/nle/?file=kop4.php` for a set of resources including ANERGazet and ANERcorp.

[34] See `http://press.jrc.it/overview.html`.

for an obligatory word boundary (white space, possibly with punctuation), "+" for one or more elements, and "*" for zero or more elements. For example, consider the rule:

Organization_BEG^{+}\b known_Name\b name_Infix*
\b Known_Name*\b Organization_END

This rule recognizes complex company names such as شركة محمد أبو المجد وإخوانه (*company of Mohamed Abu Al-Majd and Brothers*), which include person (known) names محمد أبو المجد (*Mohamed Abu Al-Majd*) and the preceding and following organization internal evidence trigger شركة (*company*) and إخوانه (*Brothers*), respectively. The Arabic NER component is able to recognize the following NE types: person, organization, location, date, and number, as well as quotations (direct reported speech) by and about people. The system was first evaluated using a corpus built from on-line news sources from the Tunisian newspaper *Assabah* and the Lebanese newspaper *Alanwar*. The system's overall performance was calculated in terms of Precision, Recall, and F-measure, delivering results of 87.17%, 65.74%, and 74.95%, respectively. Then, the system was evaluated only for person, organization, and location using ANERcorp. The system's overall performance in terms of Precision, Recall, and F-measure was 73.39%, 62.13%, and 67.13%, respectively.

10.2 Machine Learning Systems

In the field of NER, ML algorithms have been widely used in order to determine NE tagging decisions from annotated texts that are used to generate statistical models for NE prediction. Experiments reporting ML system performance are evaluated in three dimensions: the NE type, the single/combined ML classifier (learning technique), and the inclusion/exclusion of certain features from the whole feature space. Most often these experiments use a very well defined framework and their reliance on standard corpora allows for an objective comparison of the performance of a proposed system relative to existing systems.

Much research work on ML-based Arabic NER was done by Benajiba (Benajiba, Rosso, and Benedí Ruiz 2007; Benajiba and Rosso 2007, 2008; Benajiba, Diab, and Rosso 2008a, 2008b, 2009a, 2009b; Benajiba et al. 2010), who explored different ML techniques with various combinations of features. Benajiba, Rosso, and Benedí Ruiz (2007) have developed an Arabic ME-based NER system called ANERsys 1.0. The authors have built their own linguistic resources, ANERcorp and ANERgazet.[35] Lexical, contextual, and gazetteer features are used by this system. ANERsys identifies the following NE types: person, location, organization, and miscellaneous. All the experiments are carried out within the framework of the shared task of the CONLL 2002 conference. The overall system's performance in terms of Precision, Recall, and F-measure was 63.21%, 49.04%, and 55.23%, respectively. The ANERsys 1.0 system had difficulties with detecting NEs that were composed of more than one token/word. An extension of this work is ANERsys 2.0 (Benajiba and Rosso 2007), which uses a two-step mechanism for NER: 1) detecting the start and the end points of each NE, then 2) classifying the detected NEs. A POS tagging feature was exploited to improve NE boundary detection. The overall system's performance in terms of Precision, Recall, and F-measure was 70.24%, 62.08%,

[35] For ANERcorp and ANERgazet, see `http://www1.ccls.columbia.edu/~ybenajiba/`.

and 65.91%, respectively. The performance of the classification module was very good with F-measure 83.22%, although the identification phase was poor with F-measure 72.03%.

Benajiba and Rosso (2008) have applied CRF instead of ME in an attempt to improve performance. The same four types of NEs used in ANERsys 2.0 were also used in the CRF-based system. Neither Benajiba, Rosso, and Benedí Ruiz (2007) nor Benajiba and Rosso (2007) included Arabic-specific features; all the features used were language-independent. Language-independent and Arabic-specific features were used in the CRF model, including POS tags, BPC, gazetteers, and nationality. The CRF-based system achieved best results when all the features were combined. The overall system's performance in terms of Precision, Recall, and F-measure was 86.90%, 72.77%, and 79.21%, respectively. The improvement was not only dependent on the use of the CRF model but also on the additional language-specific features, including POS and BPC.

Benajiba, Diab, and Rosso (2008a) examined the lexical, contextual, morphological, gazetteer, and shallow syntactic features of ACE data sets using the SVM classifier. The system's performance was evaluated using 5-fold cross validation. The impact of the different features is measured independently and in joint combination across different standard data sets and genres. The best system's overall performance in terms of F-measure was 82.71% for ACE 2003, 76.43% for ACE 2004, and 81.47% for ACE 2005, respectively.

Benajiba, Diab, and Rosso (2008b) investigated the sensitivity of different NE types to various types of features rather than adopting a single set of features for all NE types simultaneously. The set of features examined were the lexical, contextual, morphological, gazetteer, and shallow syntactic features, forming 16 specific features in total. A multiple classifier approach was developed using SVM and CRF models, where each classifier tags an NE type separately. They used a voting scheme to rank the features according to the best performance of the two models for each NE type. The result in tagging a word with different NE types is resolved by selecting the classifier output with the highest Precision (i.e., overriding the tagging of the classifier that returned more relevant results than irrelevant). An incremental feature selection method was used to select an optimized feature set and to better understand the resulting errors. A global NER system could be developed from the union of all the optimized set of features for each NE type. ACE data sets are used in the evaluation process. The best system's overall performance in terms of F-measure was 83.5% for ACE 2003, 76.7% for ACE 2004, and 81.31% for ACE 2005, respectively. On the basis of the analysis of the best recognition results obtained by individual and combined features experiments, it cannot be concluded whether CRF is better than SVM or vice versa. Each NE type is sensitive to different features and each feature plays a role in recognizing the NE to varying degrees.

Further studies conducted in Benajiba, Diab, and Rosso (2009a, 2009b) have confirmed the importance of considering both language-independent and Arabic-specific features in the NER system. In particular, Benajiba, Diab, and Rosso (2009a) studied the impact of SVM, ME, and CRF models using the same approach and features described in Benajiba, Diab, and Rosso (2008b). The best system's overall performance in terms of F-measure was 83.34% for ACE 2003, 77.61% for ACE 2004, and 82.02% for ACE 2005, respectively. Interesting conclusions and recommendations have been suggested by this study. Both SVMs and CRFs achieved very similar performance, outperforming the ME model. An important observation concerns the number of available features as the main factor for the choice of using SVMs versus CRFs: SVMs seem to achieve good results with fewer features. Another significant observation concerns the better performance achieved by carrying out pre-processing of the Arabic text by a clitic segmenter, which is more suitable given the morphological richness of Arabic.

Later, in Benajiba et al. (2010), the Arabic NER system described in Benajiba, Diab, and Rosso (2008b) is used as a baseline NER system to automatically tag an Arabic–English parallel corpus in order to provide sufficient training data for studying the impact of deep syntactic features, also referred to as syntagmatic features. The feature space is enhanced by syntagmatic features that are bootstrapped by prediction from this corpus. These features are derived from Arabic sentence parses that include an NE. The relatively low performance of the available Arabic parser leads to noisy features as well. The inclusion of the extra features has achieved high performance for the ACE (2003–2005) data sets. The best system's overall performance in terms of F-measure was 84.32% for ACE 2003, 78.12% for ACE 2004, and 81.73% for ACE 2005, respectively. Moreover, the authors reported an F-measure improvement of up to 1.64 percentage points compared to the performance when the syntagmatic features were excluded.

Abdul-Hamid and Darwish (2010) developed a CRF-based Arabic NER system that explores using a set of simplified features for recognizing the three classic NE types: person, location, and organization. The proposed set of features include: boundary character *n*-grams (leading and trailing character *n*-gram features), word *n*-gram probability-based features that attempt to capture the distribution of NEs in text, word sequence features, and word length. Remarkably, the system did not use any external lexical resources. Moreover, the character *n*-gram models attempt to capture surface clues that would indicate the presence or absence of an NE. For example, character bigram, trigram, and 4-gram models can be used to capture the prefix attachment of a noun for a candidate NE such as the determiner الـ (*Al*), a coordinating conjunction and a determiner والـ (*w*+*Al*), and a coordinating conjunction, a preposition, and a determiner وبالـ (*w*+*b*+*Al*), respectively. On the other hand, these features can also be used to conclude that a word may not be an NE if the word is a verb that starts with any of the verb present tense character set (i.e., أ (*A*), نـ (*n*), يـ (*y*), or تـ (*t*). Despite the fact that lexical features have solved the problem of dealing with a large number of prefixes and suffixes, they do not resolve the compatibility problem between prefixes, suffixes, and stems. The compatibility checking is needed in order to verify whether a correct combination is met (cf. Buckwalter 2002). The system was evaluated using ANERcorp and the ACE 2005 data set. The overall system's performance using ANERcorp for Precision, Recall, and F-measure was 89%, 74%, and 81%, respectively. These results show that the system outperforms the CRF-based NER system of Benajiba and Rosso (2008).

Farber et al. (2008) proposed integrating a morphological-based tagger with an Arabic NER system. The integration is aimed at enhancing Arabic NER. The rich morphological information produced by MADA provides important features for the classifier. The system adopts the structured perceptron approach proposed by Collins (2002) as a baseline for Arabic NER, using morphological features produced by MADA. The system was developed to extract person, organization, and GPEs. The empirical results from a 5-fold cross validation experiment show that the disambiguated morphological features in conjunction with a capitalization feature improve the performance of the Arabic NER system. They reported 71.5% F-measure on the ACE 2005 data set.

An integrated approach was investigated in AbdelRahman et al. (2010) by combining bootstrapping, semi-supervised pattern recognition, and CRF. The feature set is extracted by the Research and Development International[36] toolkit, which includes ArabTagger and an Arabic lexical semantic analyzer. The features used include word-level, POS tag, BPC, gazetteers, semantic field tag, and morphological features. The semantic field tag is a generic cluster that refers to a set of related lexical triggers. For example, the "Corporation" cluster includes the following internal evidence that can be used to identify an organization name: مجموعة (*group*), مؤسسة (*foundation*), هيئة (*authority*), and شركة (company). The system identifies the following NEs: person, location, organization, job, device, car, cell phone, currency, date, and time. A 6-fold cross validation experiment using the ANERcorp data set showed that the system yielded F-measures of 74.06%, 89.09%, 75.01%, 69.47%, 77.52%, 80.95%, 80.63%, 98.52%, 76.99%, and 96.05% for the person, location, organization, job, device, car, cell phone, currency, date, and time NEs, respectively. The results also showed that the system outperforms the NER component of LingPipe when both are applied to the ANERcorp data set.

Mohit et al. (2012) proposed a learning (Recall-oriented) model for Arabic NER from diverse text domains like Wikipedia within the AQMAR (American and Qatari Modeling of Arabic) project. They used a flexible annotation scheme that allows for the introduction of new NE tags. As Arabic Wikipedia is not tagged for NEs, they adopted semi-supervised learning (self-training) for building their own corpus. The learning method does not utilize any gazetteer. Once the evaluation corpus is built, a supervised learning method can be used to develop and evaluate an NER classifier. The feature space consists of 15 contextual and lexical features capturing local context and shallow morphology. Morphological features are extracted from MADA output. The training model is built using the structured perceptron described in Collins (2002). This framework allows them to manipulate two key elements of the model: the features and the loss function used in training. This function measures the recognition error for each token/word, which is the difference between the correct and predicted label. It penalizes Recall errors (i.e., reduction of false negatives that arise by erroneously predicting the non-entity token/word as part of the actual NE), which is the chief difficulty for the news-text–trained model in the news domain. The system was tested on 24 Wikipedia articles[37] for possible combinations of the supervised learning phase with self-training on unlabeled Wikipedia data. The experimental results showed improvements on F-measure by the proposed Recall-oriented model in both stages of learning. When Recall-oriented bias is used in the supervised phase, the recall gains are substantial: nearly 8% over the baseline. Integrating this bias within self-training produces a more modest improvement of about 4% relative to the baseline. In both cases, the improvements to recall more than compensate for the degradation in Precision.

Zayed and El-Beltagy (2012) proposed a person NER system that automatically generates dictionaries of male and female first names as well as family names by a preprocessing step. It relies on ASVMTools (Diab, Hacioglu, and Jurafsky 2004) for POS tagging to identify proper nouns. Thereafter, the dictionaries are expanded using Web sites listing Arabic given names. The system takes into consideration the common prefixes of person names. For example, a name may take a prefix such as ال (AL, *the*), أبو (Abu, *father of*), بن (Bin, *son of*), or عبد (Abd, *servant of*), or a combination of prefixes such as أبو عبد (Abu Abd, *father of servant of*). It also takes into consideration the common embedded words in compound names. For example the person names نور الدين (*Nour Al-dain*) or

[36] http://www.rdi-eg.com/.

[37] A small corpus of Arabic Wikipedia articles was developed via a flexible entity annotation scheme spanning four topical domains (history, technology, science, and sports); this is publicly available at http://www.ark.cs.cmu.edu/AQMAR.

شمس الدين (*Shams Al-dain*) have الدين (*Al-dain*) as an embedded word. The ambiguity of having a person name as a non-NE in the text is resolved by heuristic disambiguation rules. The system is evaluated on two data sets: MSA data sets collected from news Web sites and colloquial Arabic data sets collected from the Google Moderator page. The overall system's performance using an MSA test set collected from news Web sites for Precision, Recall, and F-measure was 93.52%, 87.89%, and 90.62%, respectively. In comparison, the overall system's performance obtained using a colloquial Arabic test set collected from the Google Moderator page for Precision, Recall, and F-measure was 88.7%, 85.56%, and 87.1%, respectively.

Koulali, Meziane, and Abdelouafi (2012) developed an Arabic NER using a combined pattern extractor (a set of regular expressions) and SVM classifier that learns patterns from POS tagged text. The system covers the NE types used in the CoNLL conference, and uses a set of dependent and independent language features. Arabic features include: a determiner ال (*AL*) feature that appears as the first letters of organization names (e.g., اليونسكو, *UNESCO*) and last name (e.g., عبد الرحمن الأبنودي, *Abd Al-Rahman Al-Abnudi*), a character-based feature that denotes common prefixes of nouns, a POS feature, and a "verb around" feature that denotes the presence of an NE if it is preceded or followed by a certain verb. The system was trained on 90% of the ANERCorp data and tested on the remainder. The system was tested with different feature combinations and the best result for an overall average F-measure was 83.20%.

Bidhend, Minaei-Bidgoli, and Jouzi (2012) presented a CRF-based NER system, called Noor, that extracts person names from religious texts. Corpora of ancient religious text called NoorCorp were developed, consisting of three genres: historic, Prophet Mohammed's Hadith, and jurisprudence books. Noor-Gazet, a gazetteer of religious person names, was also developed. Person names were tokenized by a pre-processing step; for example, the tokenization of the full name حسن بن علي بن عبد الله بن المغيرة (*Hassan bin Ali bin Abd-Allah bin Al-Moghayrah*) produces six tokens as follows: حسن بن علي عبد الله المغيرة (*Hassan bin Ali Abd-Allah Al-Moghayrah*). Another pre-processing tool, AMIRA, was used for POS tagging. The tagging is enriched by indicating the presence of the person NE entry, if any, in Noor-Gazet. Details of the experimental setting are not provided. The F-measure for the overall system's performance using new historic, Hadith, and jurisprudence corpora was 99.93%, 93.86%, and 75.86%, respectively.

10.3 Hybrid Systems

The hybrid approach integrates the rule-based approach with the ML-based approach in order to optimize overall performance (Petasis et al. 2001). Recently, Abdallah, Shaalan, and Shoaib (2012) proposed a hybrid NER system for Arabic. The rule-based component is a re-implementation of the NERA system (Shaalan and Raza 2008) using GATE. The ML-based component uses Decision Trees. The feature space includes the NE tags predicted by the rule-based component and other language independent and Arabic specific features. The system identifies the following types of NEs: person, location, and organization. The F-measure performance using ANERcorp was 92.8%, 87.39%, and 86.12% for the person, location, and organization NEs, respectively.

Continuing the research of Abdallah, Shaalan, and Shoaib (2012), the hybrid Arabic NER system is extended in the following key directions (Oudah and Shaalan 2012, 2013): 1) increasing the NEs to 11 types by adding time, measurement, phone number, filename, date, price, percent, and ISBN; 2) investigating two more ML models: SVMs and Logistic Regression; and 3) increasing the features to a larger set by adding morphological features and an English-gloss capitalization feature. The experimental results showed that the hybrid Arabic NER approach outperforms the rule-based and the ML-based components when they are processed individually. The performance obtained using ANERcorp for F-measure was 94.4% for person, 90.1% for location, and 88.2% for organization NEs.

11. Conclusion

NER is one of the most fundamental and important tasks for developing NLP systems. Accurate identification of NEs from the text plays an important role for a range of NLP systems such as machine translation and information retrieval. The literature demonstrates that explicitly devoting one step of processing to NE identification helps such systems achieve better performance levels.

There are an increasing number of Arabic textual information resources available on electronic media, such as Web pages, blogs, e-mails, and text messages, which makes automated NER for the Arabic text relevant. In this survey we have presented various challenges to processing Arabic NEs, including highly ambiguous Arabic words, the absence of rigorous standards of written text, and the current state-of-the-art in Arabic NLP resources and tools.

Advances in human language technology require an ever increasing amount of data and annotation. The number of current state-of-the-art of Arabic linguistic resources is still insufficient compared with Arabic's actual importance as a language. Many existing Arabic NER resources are annotated manually or are only available at significant expense. We have described some research that adopted semi-automatic (bootstrapping) methods in order to enrich Arabic NER resources from diverse text types such as Web sources and (multilingual) corpora developed within evaluation projects. In the Arabic NER field, NEs falling under proper names representing person, location, and organization names are commonly applied to newswire domains, reflecting the importance of these limited NEs in this domain.

We have described three main approaches that have been used to develop Arabic NER systems: linguistic rule-based, ML-based, and hybrid approaches. Rule-based systems follow a classical approach and ML-based systems follow a modern and rapidly growing approach. The main reasons for choosing the rule-based approach are the lack and limitations of Arabic linguistic resources, optimized platform architectures for rule-based systems, and the high performance of such systems. In addition, ML-based approaches have proven their usefulness as they take advantage of ML algorithms by building models that include learning patterns associated with individual entity types trained from annotated data. The success of both the rule-based and ML-based approaches motivates the investigation of a hybrid Arabic NER approach, yielding significant improvements by exploiting the rule-based decisions on NEs as features used by the ML classifier.

Features are a critical aspect and are the key component for enhancing the performance of NER systems. We reviewed many attempts to select features that investigate the sensitivity of each entity when applied to different sets of features. We showed how researchers applied different techniques that benefit differently from the enabled features and obtain different results for varying NE types. Some suggest that NER for Arabic use not only language-independent features but also Arabic-specific features. Researchers sometimes exploit language-independent features based on promising variables, such as lexical and orthographic features, to overcome the problems related to the Arabic language and orthography. Lexical features avoid complex morphology by extracting the word prefix and suffix sequence of a word from the character *n*-gram of leading and trailing letters. Orthographic features attempt to overcome the lack of capitalization for NEs in Arabic by relying on the corresponding English capitalization of NEs. Alternatively, other researchers suggest including a rich set of language specific features extracted by Arabic morpho-syntactic tools in order to deeply analyze the inherent complex structure of NEs within their context. Regardless of the features selected, various studies have reported that significant system performance is achieved when a combination that includes all features is enabled.

We have discussed many existing tools that have been used to build many different Arabic NER systems. IDEs are convenient for rapid development of NER systems. GATE is more diversified and comprehensive for developing rule-based Arabic NER systems because it has built-in gazetteers and rules offering the ability to create new ones. On the other hand, the availability of diverse generic ML tools is sufficient for developing a wide range of Arabic NER classifiers. The main problem with these generic tools is that they are language-independent with limited support for Arabic. Fortunately, the availability of Arabic morpho-syntactic pre-processing tools, such as BAMA and its successor MADA for morphological processing and AMIRA for BPC, has lessened the need for extensive development efforts.

Almost all of the tools adopted for developing Arabic NER provide for system evaluation by calculating the value of Precision, Recall, and F-measure. Sometimes it is too expensive to acquire linguistic evaluation resources to compare a proposed system's performance to existing systems. Fortunately, the increasing contributions from the Arabic NLP research community have been sufficient to provide a practical solution and satisfy the critical need for free corpora and gazetteers (e.g., ANERsys, which can be used to compare Arabic NER under different experimental settings).

We have reviewed the state-of-the-art in Arabic NER systems in some detail. It should be noted that the list of references provided here may not be comprehensive. Our aim was to provide a review of the essential aspects of Arabic NER and discuss major publications that have made use of those ideas. We hope that this survey provides a way to access the main branches of the literature dealing with Arabic NER research and guides researchers in interesting and fruitful research directions.

Since the presence of NE in the context of one language points to a correspondence in other natural languages, studies of NEs in one language could provide mutual and valuable insight for developing resources and technologies that can handle NEs in many languages. This survey describes the progress made by Arabic NER research. This study might be easily extrapolated to most NLP tasks in general and to many of the morphologically complex/rich languages in particular.

References

Abdallah, Sherief, Khaled Shaalan, and Muhammad Shoaib. 2012. Integrating rule-based system with classification for Arabic named entity recognition. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 7181 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, pages 311–322.

Abdel Monem, Azza, Khaled Shaalan, Ahmed Rafea, and Hoda Baraka. 2008. Generating Arabic text in multilingual speech-to-speech machine translation framework. *Machine Translation*, 22(4):205–258.

AbdelRahman, Samir, Mohamed Elarnaoty, Marwa Magdy, and Aly Fahmy. 2010. Integrated machine learning techniques for Arabic named entity recognition. *International Journal of Computer Science Issues (IJCSI)*, 7(4):27–368.

Abdul-Hamid, Ahmed and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop (NEWS 2010)*, pages 110–115, Stroudsburg, PA.

Abdul-Mageed, Muhammad, Mona Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard Arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT 2011): short papers - Volume 2*, pages 587–591, Stroudsburg, PA.

Abouenour, Lahsen, Karim Bouzoubaa, and Paolo Rosso. 2010. Using the yago ontology as a resource for the enrichment of named entities in Arabic wordnet. In *Proceedings of The Seventh International Conference on Language Resources and Evaluation (LREC 2010) Workshop on Language Resources and Human Language Technology for Semitic Languages*, pages 27–31, Valletta.

Abuleil, Saleem. 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 (RIAO 2004)*, pages 638–647, Vaucluse.

Al-Jumaily, Harith, Paloma Martínez, Martínez-Fernández José, and Erik Goot. 2012. A real time named entity recognition system for Arabic text mining. *Language Resources and Evaluation Journal*, 46(4):543–563.

Al-Onaizan, Yaser and Kevin Knight. 2002a. Machine transliteration of names in Arabic text. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages (SEMITIC 2002)*, pages 1–13, Stroudsburg, PA.

Al-Onaizan, Yaser and Kevin Knight. 2002b. Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 400–408, Stroudsburg, PA.

Al-Shalabi, Riyad, Ghassan Kanaan, Bashar Al-Sarayreh, Khalid Khanfar, Ali AIGhonmein, Hamed Talhouni, and Salem Al-Azazmeh. 2009. Proper noun extracting algorithm for the Arabic language. In *International Conference on IT to Celebrate S. Charmonman's 72nd Birthday*, pages 28.1–28.9, Bangkok.

Al-Sughaiyer, Imad and Ibrahim Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Algahtani, Shabib. 2011. *Arabic Named Entity Recognition: A Corpus-Based Study*. Ph.D. thesis, The University of Manchester, UK.

Alkharashi, Ibrahim. 2009. Person named entity generation and recognition for Arabic language. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 205–208, Cairo.

Attia, Mohammed, Antonio Toral, Lamia Tounsi, Monica Monachini, and Josef van Genabith. 2010. An automatically built named entity lexicon for Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 3,614–3,621, Valletta.

Babych, Bogdan and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, EAMT 2003, pages 1–8, Stroudsburg, PA.

Badawy, Osama, Mohamed Shaheen, and Abdelbaki Hamadene. 2011. ARQA: An intelligent Arabic question answering system. In *Proceedings of Arabic Language Technology International Conference (ALTIC 2011)*, pages 1–8, Alexandria.

Balasuriya, Dominic, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, People's Web 2009, pages 10–18, Stroudsburg, PA.

Ben Hamadou, Abdelmajid, Piton Odile, and Fehri Héla. 2010a. Multilingual extraction of functional relations between Arabic named entities using NooJ platform. In *HAL Archives*, pages 1–10, Available at `http://hal.archives-ouvertes.fr/hal-00547940`.

Ben Hamadou, Abdelmajid, Piton Odile, and Fehri Héla. 2010b. Recognition and Arabic-French translation of named entities: Case of the sport places. In *arXiv*, pages 1–10, Available at `https://www.researchgate.net/publication/45898820_Recognition_and_translation_Arabic-French_of_Named_Entities_case_of_the_Sport_places`.

Benajiba, Yassine, Mona Diab, and Paolo Rosso. 2008a. Arabic named entity recognition: An SVM-based approach. In *Proceedings of Arab International Conference on Information Technology (ACIT 2008)*, pages 16–18, Hammamet.

Benajiba, Yassine, Mona Diab, and Paolo Rosso. 2008b. Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 284–293, Stroudsburg, PA.

Benajiba, Yassine, Mona Diab, and Paolo Rosso. 2009a. Arabic named entity recognition: A feature-driven study. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):926–934.

Benajiba, Yassine, Mona Diab, and Paolo Rosso. 2009b. Using language independent and language specific features to enhance Arabic named entity recognition. *The International Arab Journal of Information Technology (IAJIT)*, 6(5):463–471.

Benajiba, Yassine and Paolo Rosso. 2007. ANERsys 2.0: Conquering the NER task for the Arabic language by combining the maximum entropy with POS-tag information. In *Proceedings of Workshop on Natural Language-Independent Engineering, 3rd Indian International Conference on Artificial Intelligence (IICAI-2007)*, pages 1,814–1,823, Mumbay.

Benajiba, Yassine and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proceedings of the Workshop on HLT & NLP within the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 143–153, Marrakech.

Benajiba, Yassine, Paolo Rosso, and José Miguel Benedí Ruiz. 2007. ANERsys: An Arabic named entity recognition system based on maximum entropy. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2007)*, pages 143–153, Berlin.

Benajiba, Yassine, Imed Zitouni, Mona Diab, and Paolo Rosso. 2010. Arabic named entity recognition: Using features extracted from noisy data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort 2010, pages 281–285, Stroudsburg, PA.

Bidhend, Majidi, Behrouz Minaei-Bidgoli, and Hosein Jouzi. 2012. Extracting person names from ancient Islamic Arabic texts. In *Proceedings of Language Resources and Evaluation for Religious Texts (LRE-Rel) Workshop Programme, Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1–6, Istanbul.

Bies, Ann, Denise DiPersio, and Mohamed Maamouri. 2012. Linguistic resources for Arabic machine translation: The Linguistic Data Consortium (LDC) catalog. In Abdelhadi Soudi, Ali Farghaly, Günter Neumann, and Rabih Zbib, editors, *Challenges for Arabic Machine Translation*, volume 322 of *Natural Language Processing 9*. John Benjamins Publishing Company, Amesterdam, pages 15–22.

Brini, Wissal, Mariem Ellouze, Omar Trigui, Slim Mesfar, Lamia Hadrich, and Paolo Rosso. 2009. Factoid and definitional Arabic question answering system. In *Proceedings of the NOOJ-2009*, pages 1–11, Tozeur.

Buckwalter, Tim. 2002. Buckwalter Arabic morphological analyzer version 1.0. Technical Report LDC2002L49, Linguistic Data Consortium (LDC), Philadelphia, PA.

Burkett, David, Slav Petrov, John Blitzer, and Dan Klein. 2010. Learning better monolingual models with unannotated bilingual text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL 2010, pages 46–54, Stroudsburg, PA.

Chen, Hsin-Hsi, Changhua Yang, and Ying Lin. 2003. Learning formulation and transformation rules for multilingual named entities. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 1–8, Boston, MA.

Collins, Michael. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP 2002, pages 1–8, Stroudsburg, PA.

Cunningham, Hamish. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*. GATE publications.

Diab, Mona. 2009. Second generation tools (AMIRA 2.0): Fast and robust tokenization, POS tagging, and Base Phrase Chunking. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 285–288, Cairo.

Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of Human Language Technology-North American Association for Computational Linguistics*, HLT-NAACL-Short 2004, pages 149–152, Stroudsburg, PA.

El Kholy, Ahmed and Nizar Habash. 2010. Techniques for Arabic morphological detokenization and orthographic denormalization. In *Proceedings of the Workshop on Language Resources and Human Language Technology for Semitic Languages in the Language Resources and Evaluation Conference (LREC)*, pages 45–51, Valletta.

Elgibali, Alaa. 2005. *Investigating Arabic: Current Parameters in Analysis and Learning*. Studies in Semitic Languages and Linguistics Series. Brill Academic Publishers, Boston, MA.

Elkateb, Sabri, William Black, Piek Vossen, David Farwell, Adam Pease, and Christiane Fellbaum. 2006. Arabic WordNet and the challenges of Arabic. In *Proceedings of Arabic NLP/MT Conference*, pages 15–24, London.

Elsebai, Ali and Farid Meziane. 2011. Extracting persons names from Arabic newspapers. In *Proceedings of the International Conference on Innovations in Information Technology*, pages 87–89, Dubai.

Elsebai, Ali, Farid Meziane, and Fatma Belkredim. 2009. A rule based persons names Arabic extraction system. In *Proceedings of the 11th International Business Information Management Association Conference (IBIMA 2009), Special Track on Arabic Information Processing*, pages 53–59, Cairo.

Ezzeldin, Ahmed and Mohamed Shaheen. 2012. A survey of Arabic question answering: Challenges, tasks, approaches, tools, and future trends. In *Proceedings of The 13th International Arab Conference on Information Technology (ACIT 2012)*, pages 1–8, Zarqa.

Farber, Benjamin, Dayne Freitag, Nizar Habash, and Owen Rambow. 2008. Improving NER in Arabic using a morphological tagger. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2,509–2,514, Marrakech.

Farghaly, Ali and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):1–22.

Fellbaum, Christiane. 2005. Wordnet and wordnets. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*. Oxford, Elsevier, pages 665–670.

Grefenstette, Gregory, Nasredine Semmar, and Faïza Elkateb-Gara. 2005. Modifying a natural language processing system for European languages to treat Arabic in information processing and information retrieval applications. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 31–38, Ann Arbor, MI.

Guo, Jiafeng, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 267–274, New York City.

Habash, Nizar. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers.

Habash, Nizar and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, ACL 2005, pages 573–580, Stroudsburg, PA.

Habash, Nizar, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo.

Halpern, Jack. 2009. Lexicon-driven approach to the recognition of Arabic named entities. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 193–198, Cairo.

Hassan, Ahmed, Haytham Fahmy, and Hany Hassan. 2007. Improving named entity translation by exploiting comparable and parallel corpora.

In *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, pages 1–6, Borovets.

Hassan, Hany and Jeffrey Sorensen. 2005. An integrated approach for Arabic-English named entity translation. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 87–93, Ann Arbor, MI.

Hewavitharana, Sanjika and Stephan Vogel. 2011. Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora, 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 61–68, Portland, OR.

Higgins, Chiara, Elizabeth McGrath, and Lailla Moretto. 2010. Mturk crowdsourcing: A viable method for rapid discovery of Arabic nicknames? In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 89–92, Stroudsburg, PA.

Huang, Shudong, Stephanie Strassel, Alexis Mitchell, and Zhiyi Song. 2004. Shared resources for multilingual information extraction and challenges in named entity annotation. In *Proceedings of the IJCNLP-04 Workshop on Named Entity Recognition for NLP Applications*, pages 112–119, Hainan Island.

Kim, SeonYeong, Sung-Hwan Kim, and Hwan-Gue Cho. 2012. Developing a system for searching a shop name on a mobile device using voice recognition and GPS information. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ICUIMC 2012)*, pages 1–8, New York City.

Korayem, Mohammed, David Crandall, and Muhammad Abdul-Mageed. 2012. Subjectivity and sentiment analysis of Arabic: A survey. In Aboul-Ella Hassanien, Abdel-Badeeh M. Salem, Rabie Ramadan, and Tai-hoon Kim, editors, *Advanced Machine Learning Technologies and Applications*, volume 322 of *Communications in Computer and Information Science*. Springer, Berlin Heidelberg, pages 128–139.

Koulali, Rim, Meziane, and Abdelouafi. 2012. A contribution to Arabic named entity recognition. In *Proceedings of the 10th International Conference on ICT and Knowledge Engineering*, pages 46–52, Morocco.

Kumaran, A., Mitesh M. Khapra, and Haizhou Li. 2010. Report of NEWS 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, NEWS 2010, pages 21–28, Stroudsburg, PA.

Lahsen, Abouenour, Karim Bouzoubaa, and Paolo Rosso. 2012. IDRAAQ: New Arabic question answering system based on query expansion and passage retrieval. In *Online Working Notes/Labs/Workshop of the CLEF 2012, Question Answering for Machine Reading Evaluation (QA4MRE) main task*, Rome, Italy.

Ma, Xiaoyi. 2010. Toward a name entity aligned bilingual corpus. In *Proceedings of LREC 2010 Workshop on Methods for the Automatic Acquisition of Language Resources and their Evaluation Methods*, pages 211–216, Valletta.

Maamouri, Mohamed, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic treebank: Building a large-scale annotated Arabic corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo.

Maloney, John and Michael Niv. 1998. TAGARAB: A fast, accurate Arabic name recognizer using high-precision morphological analysis. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, Semitic 1998, pages 8–15, Stroudsburg, PA.

Marton, Yuval, Nizar Habash, and Owen Rambow. 2010. Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, SPMRL '10, pages 13–21, Stroudsburg, PA.

Maynard, Diana, Hamish Cunningham, Kalina Bontcheva, Roberta Catizone, George Demetriou, Gaizauskas Robert, Oana Hamza, Mark Hepple, Patrick Herring, Brian Mitchell, Michael Oakes, Wim Peters, Andrea Setzer, Mark Stevenson, Valentin Tablan, Christian Ursu, and Yorick Wilks. 2000. A survey of uses of gate. Technical Report CS-00-06, Department of Computer Science, University of Sheffield.

Maynard, Diana, Hamish Cunningham, Kalina Bontcheva, and Marin Dimitrov. 2002. Adapting a robust multi-genre NE system for automatic content extraction. In Donia Scott, editor, *Artificial Intelligence: Methodology, Systems, and Applications*, *10th International Conference, Varna, Bulgaria*, volume 2443 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, pages 264–273.

Mesfar, Slim. 2007. Named entity recognition for Arabic using syntactic grammars. In *Proceedings of the 12th International Conference on Applications of Natural Language to Information Systems (NLDB'07)*, pages 305–316, Berlin.

Mohit, Behrang, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah Smith. 2012. Recall-oriented learning of named entities in Arabic wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2012*, pages 162–173, Stroudsburg, PA.

Mollá, Diego, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In Zakerman Covendon, Lowrence and Ingrid, editors, *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW 2006)*, pages 51–58, Sydney.

Mostefa, Djamel, Stéphane Chaudiron, Laïb Mariama, Khalid Choukri, and Gaël de Chalendar. 2009. A multilingual named entities corpus for Arabic, English and French. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 213–216, Cairo.

Nadeau, David and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Nezda, Luke, Andrew Hickl, John Lehmann, and Sarmad Fayyaz. 2006. What in the world is a shahab? Wide coverage named entity recognition for Arabic. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-06)*, pages 41–46, Genoa.

Oudah, Mai and Khaled Shaalan. 2012. A pipeline Arabic named entity recognition using a hybrid approach. In *Proceedings of the International Conference on Computational Linguistics*, pages 2,159–2,176, Mumbai.

Oudah, Mai and Khaled Shaalan. 2013. Person name recognition using the hybrid approach. In Elisabeth Métais, Farid Meziane, Mohamad Saraee, Vijayan Sugumaran, and Sunil Vadera, editors, *Natural Language Processing and Information Systems*, volume 7934 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, pages 237–248.

Pappu, Aasish. 2009. Using wikipedia for hierarchical finer categorization of named entities. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 779–786, Hong Kong.

Petasis, Georgios, Frantz Vichot, Francis Wolinski, Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos. 2001. Using machine learning to maintain rule-based named-entity recognition and classification systems. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 426–433, Stroudsburg, PA.

Pouliquen, Bruno, Ralf Steinberger, Camelia Ignat, Irina Temnikova, Anna Widiger, Wajdi Zaghouani, and Jan Zizka. 2005. Multilingual person name recognition and transliteration. In *arXiv*, pages 1–10. Available at `http://www.researchgate.net/publication/1959893_Multilingual_person_name_recognition_and_transliteration/file/d912f50ecfc2949c6d.pdf`.

Ratinov, Lev and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, CO.

Refaat, Khaled and Amgad Madkour. 2009. An optimized method for Arabic cross document named entity normalization. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 209–212, Cairo.

Ryding, Karin. 2005. *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press, New York.

Salloum, Wael and Nizar Habash. 2012. Elissa: A dialectal to standard Arabic machine translation system. In *Proceedings of the International Conference on Computational Linguistics: Demonstration Papers*, pages 385–392, Mumbai.

Samy, Doaa, Antonio Moreno, and José Guirao. 2005. A proposal for an Arabic named entity tagger leveraging a parallel corpus. In *Proceedings of the 2005 Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 459–465, Borovets.

Saravanan, K., Monojit Choudhury, Raghavendra Udupa, and A. Kumaran. 2012. An empirical study of the occurrence and co-occurrence of named entities in natural language corpora. In *Proceedings* *of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3,118–3,125, Istanbul.

Sekine, Satoshi, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1,818–1,824, Las Palmas.

Shaalan, Khaled. 2010. Rule-based approach in Arabic natural language processing. *The International Journal on Information and Communication Technologies (IJICT)*, 3(3):11–19.

Shaalan, Khaled, Mohammed Attia, Pavel Pecina, Younes Samih, and Josef van Genabith. 2012. Arabic word generation and modelling for spell checking. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, 719–725, Istanbul.

Shaalan, Khaled and Hafsa Raza. 2007. Person name entity recognition for Arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Semitic 2007, pages 17–24, Stroudsburg, PA.

Shaalan, Khaled and Hafsa Raza. 2008. Arabic named entity recognition from diverse text types. In Bengt Nordström and Aarne Ranta, editors, *Advances in Natural Language Processing*, volume 5221 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, pages 440–451.

Shaalan, Khaled and Hafsa Raza. 2009. NERA: Named entity recognition for Arabic. *Journal of the American Society for Information Science and Technology*, 60(8):1,652–1,663.

Shihadeh, Carolin and Günter Neumann. 2012. ARNE: A tool for named entity recognition from Arabic text. In *Fourth Workshop on Computational Approaches to Arabic Script-based Languages (CAASL4), located at the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 24–31, San Diego, CA.

Smrz, Otakar. 2007. *Functional Arabic Morphology Formal System and Implementation*. Ph.D. thesis, Charles University in Prague, Czech Republic.

Steinberger, Ralf. 2012. A survey of methods to ease the development of highly multilingual text mining applications. *Language Resources and Evaluation Journal*, 46(2):155–176.

Steinberger, Ralf, Bruno Pouliquen, and Camelia Ignat. 2008. Using language-independent rules to achieve high multilinguality in text mining. In Francois Fogelman-Soulie, Domenico Perrotta, Jakub Piskorski, and Ralf Steinberger, editors, *Mining Massive Data Sets for Security: Advances in Data Mining, Search, Social Networks and Text Mining, and their Applications to Security*, volume 19 of *Information and Communication Security*. IOS Press, Amsterdam, Netherlands, pages 217–240.

Steinberger, Ralf, Bruno Pouliquen, and Erik Van der Goot. 2009. An introduction to the Europe media monitor family of applications. In *Information Access in a Multilingual World-Proceedings of the SIGIR 2009 Workshop (SIGIR-CLIR 2009)*, pages 1–8, Boston, MA.

Strassel, Stephanie, Alexis Mitchell, and Shudong Huang. 2003. Multilingual resources for entity extraction. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-Language Named Entity Recognition - Volume 15*, MultiNER '03, pages 49–56, Stroudsburg, PA.

Traboulsi, Hayssam. 2009. Arabic named entity extraction: A local grammar-based approach. In *Proceedings of the International Multi-conference on Computer Science and Information Technology (IMCSIT 2009)*, pages 139–143, Mragowo.

Trigui, Omar, Lamia Belguith, Paolo Rosso, Hichem Ben Amor, and Bilel Gafsaoui. 2012. Arabic QA4MRE at CLEF 2012: Arabic question answering for machine reading evaluation. In *Online Working Notes/Labs/Workshop of the CLEF 2012, Question Answering for Machine Reading Evaluation (QA4MRE main task)*, Rome, Italy. CLEF.

Witten, Ian, Eibe Frank, and Mark Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science.

Yang, Yiming. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90.

Zaghouani, Wajdi. 2012. RENAR: A rule-based Arabic named entity recognition system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(1):2:1–2:13.

Zaghouani, Wajdi, Bruno Pouliquen, Mohamed Ebrahim, and Ralf Steinberger. 2010. Adapting a resource-light highly multilingual named entity recognition system to Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 563–567, Valletta.

Zaraket, Fadi and Jad Makhlouta. 2012. Arabic cross-document NLP for the hadith and biography literature. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2012)*, pages 256–261, Marco Island, FL.

Zayed, Omnia and Samhaa El-Beltagy. 2012. Person name extraction from modern standard Arabic or colloquial text. In *Proceedings of the 8th International Conference on Informatics and Systems Conference (INFOS2012)*, NLP track, pages 44–48, Cairo.

Zitouni, Imed, Jeff Sorensen, Xiaoqiang Luo, and Radu Florian. 2005. The impact of morphological stemming on Arabic mention detection and coreference resolution. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic 2005, pages 63–70, Stroudsburg, PA.