# Chapter 7
# Named Entity Recognition

**Behrang Mohit**

## 7.1 Introduction

Named entity recognition (NER) is the problem of locating and categorizing important nouns and proper nouns in a text. For example, in news stories names of persons, organizations and locations are typically important. In the following example, the highlighted named entities hold key information and are useful for language processing applications.

*Before joining **UCB**, **Lisa North** worked for **Pegasus Books** in **North Berkeley**.*

Named entity recognition plays an important role in applications such as Information Extraction, Question Answering and Machine Translation. For example, information about named entities such as *Lisa North* helps a machine translation system to avoid translating them erroneously word by word.

The NER task has been studied extensively for many languages [54] including Arabic and Hebrew. Throughout the past two decades, numerous systems and data resources have been developed for NER. Moreover, there has been several forums and evaluation programs focused on named entity recognition and other related tasks.

In this chapter, we review the general state of NER research, relevant challenges and the current state of the art works on Semitic NER. Specifically, we look into two case studies for Arabic and Hebrew named entity recognition. We also review Semitic NLP tasks which overlap with the named entity recognition. We close with an overview of the available resources for Semitic NER and some the open research questions.

B. Mohit (✉)
Carnegie Mellon University in Qatar, Doha, Qatar
e-mail: behrang@cmu.edu

I. Zitouni (ed.), *Natural Language Processing of Semitic Languages*,
Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-45358-8_7, © Springer-Verlag Berlin Heidelberg 2014

**Table 7.1** Sample NER output with the mention-level (SGML) and BIO and BIOLU representations

| Representation | Example | | |
|---|---|---|---|
| SGML | <PER>Dr. Doull</PER> from the <ORG>Royal College of Paediatrics</ORG> in <LOC>Wales</LOC> backed the <MIS>Fresh Start</MIS>. | | |
| | **Token** | **BIO** | **BIOLU** |
| BIO & BIOLU | Dr. | B-PER | B-PER |
| | Doull | I-PER | L-PER |
| | from | O | O |
| | the | O | O |
| | Royal | B-ORG | B-ORG |
| | College | I-ORG | I-ORG |
| | of | I-ORG | I-ORG |
| | Paediatrics | I-ORG | L-ORG |
| | in | O | O |
| | Wales | B-LOC | U-LOC |
| | backed | O | O |
| | the | O | O |
| | Fresh | B-MIS | B-MIS |
| | Start | I-MIS | L-MIS |
| | . | O | O |

## 7.2 The Named Entity Recognition Task

### *7.2.1 Definition*

Named entities (NEs) are words or phrases which are named or categorized in a certain topic. They usually carry key information in a sentence which serve as important targets for most language processing systems. Accurate named entity recognition can be used as a useful source of information for different NLP applications. For example the performance of applications like Question Answering [69], Machine Translation [7] or Information Retrieval [39] has been improved by named entity information. Table 7.1 shows an example sentence annotated with the named entity information, using different representation schemes. The three intuitive classes of person (PER), location (LOC), organization (ORG) along with the loosely defined miscellaneous(MIS) class are used in most NER systems. These classes are mostly relevant to the news related corpora. For other domains, NER systems are expected to be trained and tested with other relevant class labels.

Table 7.1 also presents different representations of named entity annotation. Early NER approaches used the mention (chunk) level representation which annotated a named entity as a whole chunk [66]. As the task evolved into a statistical learning problem, the sequence labeling framework became the standard

approach [16, 49]. In sequence labeling, the entire sequence of tokens (usually the sentence) is labeled concurrently. The BIO labeling is a representation that is generally used for sequence labeling. In this representation, a token is seen to be at the **B**eginning or **I**nside or **O**utside a named entity. In the alternative BILOU representation, the **L** and **U** labels are used respectively for the **L**ast token of a multi-token entity and the **U**nit-length named entities.[1]

The scope of named entity recognition has evolved over the past couple of decades. Originally NER was limited to the extraction of news related proper nouns such as names of persons, organizations and locations. With the expansion of NLP in other domains, those few traditional named entity classes were not sufficient. For example, for an article about science or technology, the three traditional classes are not enough and other named entity classes need to be considered. Moreover, named entities should not be limited to proper nouns. In certain areas of studies such as nuclear physics, one might highlight terms such as *proton* or *uranium* as named entities.[2] Thus, despite the common focus on the person, location and organization classes one can say that *NER encompasses the extraction of all important entities in a given context.*

### *7.2.2 Challenges in Named Entity Recognition*

Named entity recognition consists of the following two sub-problems: (1) recognition of named entity boundaries; (2) recognition of named entity categories (classes). These problems are usually (but not necessarily) addressed concurrently. Similar to most problems in language processing, there are ambiguities in the language which add to the challenge of the task. In the following, we present examples of ambiguities in both recognition and categorization of named entities. In the first sentence, there is an ambiguity in the recognition of the named entity *Reading* that can be confused as a gerund form of a verb or a proper noun (city name). In the second example, the ambiguity is in the named entity type; *Fox* can be interpreted either as a person, an organization or a non-named entity. Furthermore, *Washington* might refer to a person, location or organization (US. government).

- **Reading** *is located between two major highways.*
  <LOC> **Reading** </LOC> is located between two major highways.
- **Fox** *criticized* ***Washington***.
  <ORG> **FOX** </ORG> criticized <ORG>**Washington**</ORG>.

[1] Ratinov and Roth [59] have shown that with a small linear expansion of the parameters, the BILOU representation results in a better NER performance.

[2] Temporal and numerical expressions are other examples named entities which are not proper nouns.

**Table 7.2** Examples of rules used to extract named entities

| | |
|---|---|
| Pattern | "headquartered in <x>" |
| Known locations | *Nicaragua* |
| New locations | *San Miguel, Chapare region, San Miguel City* |
| Pattern | "to occupy <x>" |
| Known locations | *Nicaragua* |
| New locations | *San Sebastian neighborhood* |

Most NER challenges lie in its heavily lexicalized and domain-dependent nature. Names take a large part of a language and are constantly evolving in different domains. In order to have a robust NER system for any given domain (e.g. tourism), we need labeled corpora and lexicons (e.g. names of monuments). Creating and updating such resources for various topics is an expensive task and requires linguistics and domain expertise. In the following we will review two frameworks of *rule-based* and *statistical* NER and will discuss their data requirements and robustness.

### *7.2.3 Rule-Based Named Entity Recognition*

Early approaches to named entity recognition were primarily rule-based. Most rule-based systems used three major components: (1) a set of named entity extraction rules, (2) gazeteers[3] for different types of named entity classes, and (3) the extraction engine which applies the rules and the lexicons to the text. The rule set and the lexicons were either completely handcrafted by humans or were bootstrapped from a few hand-crafted examples. A successful example of the rule-based framework was the AutoSlog Information Extraction system [61]. Table 7.2 presents samples of Auto-Slog's rules and the extracted named entities.[4] The system starts with a set of simple seed rules for some known entities like *Nicaragua*. In an iterative bootstrapping framework the rules were applied and got extended to extract new entities like *San Sebastian*.

Rule-based systems are relatively precise but usually have low coverage and work well on narrow domains. Their performance usually depends on how comprehensive the rules and lexicons are. Bootstrapping frameworks like [61] are still limited to the domain of the seed rules and lexicon. Furthermore, incorporation of deeper knowledge beyond the surface words and lexicons in to a rule-based system requires expensive manual effort. In contrast, statistical frameworks are more flexible in incorporating richer linguistic knowledge (e.g. syntax) which results in more robust systems.

[3] *Gazeteer* is a term that is commonly used to refer to a domain specific lexicon. For example, there are gazetteers for country and city names.

[4] Example is borrowed from [60].

### 7.2.4 Statistical Named Entity Recognition

The rising popularity of the statistical NLP methods along with the expansion of available data resources has directed NER research to data-driven and statistical methods. The use of statistical methods reduced the human effort needed for the tedious construction of rule sets and gazeteers. Soon after their development, statistical and hybrid systems like [51, 52] outperformed the state of the art rule-based systems.

Statistical named entity recognition usually uses the following two main components:

1. Labeled training data: text corpora where named entities are annotated (similar to examples in Table 7.1).
2. A statistical model: a probabilistic representation of the training data.

A statistical model is made of parameters which map a language *event* to a probability. For example a statistical model that is trained on our earlier example *(Fox criticized Washington)*, might have parameters such as the probability of the first word in a sentence being a named entity or the probability of certain word (e.g. *Fox*) being labeled as *organization*.

As a supervised learning problem, named entity recognition can be modeled as a classification task for each individual token. However, such approach fails to consider the interdependency between different tokens. In contrast, NER is usually seen as a *structured learning* problem for a sequence of variables. That is the *sequence labeling* view where the learner predicts the labels for the entire sequence of tokens (usually a sentence). This approach allows the modeling of the dependency that exists between different tokens. For example in the earlier example, the class disambiguation for the word *Fox* is easier if the entire sequence (specially the word *Washington*) are included in the prediction.

In a sequence labeling framework a sentence is represented by a set of token variables $t_1, t_2, \ldots, t_N$. The labeler is expected to find the most likely sequence of named entity labels, $y_1, y_2, \ldots, y_N$. The set of labels consists of the BIO boundaries along with the named entity types. Thus, the class possibilities for a model which labels person, location, organization are: B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG and O.

Formulating the problem probabilistically, we would like to find the label sequence which satisfies:

$$S = \underset{y_1 \ldots y_N}{\operatorname{argmax}} P(y_1 \ldots y_N | t_1 \ldots t_N) \tag{7.1}$$

Using the Bayes' theorem of probabilities, we can rewrite and simplify the above formula as:

$$S = \underset{y_1 \ldots y_N}{\operatorname{argmax}} P(t_1 \ldots t_N | y_1 \ldots y_N) P(y_1 \ldots y_N) \tag{7.2}$$

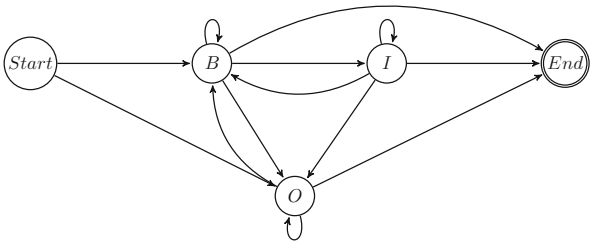


**Fig. 7.1** An simplified HMM for detect NE boundaries

There are different ways of modeling the sequence labeling problem. One well-known approach is the hidden Markov model (HMM) [58]. HMM is based on two concepts:

1. A probabilistic graphical model in which class variables are represented by states which are able to *generate* tokens.
2. An assumption that there is a Markov process in the generation of the tokens. The assumption is that the probability of assigning a class to a token depends only on a few earlier tokens (and their class labels).

HMM formulates the labeling problem as:

$$S = \underset{y_1 \ldots y_N}{\text{argmax}}\, P(t_1 \ldots t_n | y_1 \ldots y_n) P(y_1 \ldots y_n) \tag{7.3}$$

$$= \prod_{i=1,\ldots,N} P(t_i | y_i) P(y_i | y_{i-1}) \tag{7.4}$$

In the formulation shown above, the Markov assumption allows us to shorten the context for computing $P(y_1 \ldots y_n)$ and simply use $P(y_i | y_{i-1})$. This is the first order HMM in which the model includes the contextual information for one previous word. Richer models with higher order use longer context with much larger parameter space.

Figure 7.1 presents an HMM for a simplified task of finding named entity boundaries. In this model, the class labels are limited to only three boundary labels (B, I and, O). The start and the end states are used to enforce boundaries for the sequence labeling task. Here, the sequence labeling of named entity boundaries follows a *generative* story:

1. The sequence begins at the *Start* state.
2. For each token position in the sequence, there is a probabilistic state transition where the class label gets decided.
3. After each transition, the destination state *generates* a word.
4. The sequence finishes at the *End* state.

In order to follow the above HMM framework, two sets of parameters are needed to train the HMM:

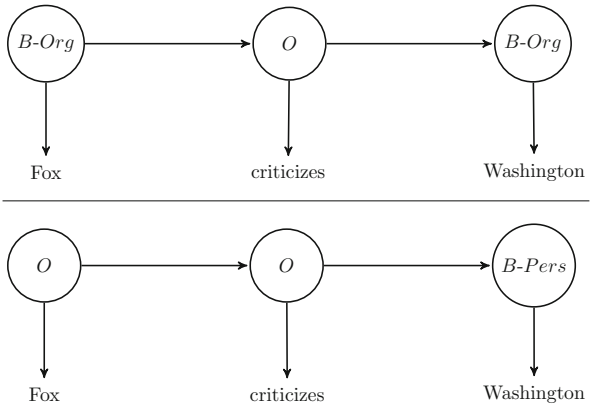


**Fig. 7.2** An ambiguous example with the correct and an incorrect labeling by HMM

1. $P(y_i|y_{i-1})$: state transition probability which is the conditional probability of the current token's label given the previous token's label.
2. $P(t_i|y_i)$: the probability of generating a token, given its label.

During the training, the model learns these two sets of parameters by counting and calculating the probability of different state transitions and word generations in the training data.

Having a trained HMM, we can choose the most likely tag sequence that maximizes the product the two parameters. Since the labeling takes place globally for the entire sequence, the model can deal with some of the class ambiguities. Figure 7.2 presents the correct and an incorrect sequence of HMM states (labels) for an ambiguous sequence. Here, the tagging of *Fox* as (news) organization influences the following state sequence and results in the tagging of *Washington* as a (government) organization. In the second labeling, the model collectively labels *Fox* as non-NE and *Washington* as person.

In general, the procedure to find the most likely label (state) sequence is named *decoding*. Methods such as the Viterbi algorithm which use dynamic programming, are commonly used for the HMM decoding.[5]

In order to train richer NER models, one would like to incorporate deeper linguistic information like long distance dependencies, morphological agreements, etc. HMM assumes that tokens are independent of each other. This assumption limits the scope of the contextual information that the NER model can use. Thus, learning features are limited to the current token [16].

In richer discriminative models such as the Maximum Entropy [15], the Perceptron [20] and the Conditional Random Fields (CRF) [41], there is no assumption made about the independence of the words and their class labels. This relaxed framework allows the model to benefit from diverse overlapping (non-independent) features [13,49]. For example, the model can use different lexicons of foreign names

[5]Two well-explained usage of the above HMM framework can be found in [37,48].

or cultural genres [59]. Moreover, global features which are collected in context beyond the current sentence have also been incorporated into discriminative models [19, 59].

### 7.2.5 Hybrid Systems

Hybrid named entity recognition systems combine two or more systems to reach a collective decision. These systems have shown improvement over their baseline counterparts. The work of [17] in combining statistical and rule-based systems in the MUC competitions as well as the work of [26] in combining different statistical learning algorithms are two successful examples of hybrid NER. In Sect. 7.4 we will discuss two Semitic NER systems that use hybrid frameworks, with different learning algorithms.

### 7.2.6 Evaluation and Shared Tasks

Named entity recognition systems are evaluated by running them on human-labeled data and comparing their results against this gold-standard. The comparison is usually at the phrase level, giving full credit for complete boundary and category matches and no credit for partial matches. The commonly used evaluation metrics are the *precision* and *recall* which have been borrowed from Information Retrieval evaluation. *Recall* measures the coverage of the system i.e. the percentage of gold-standard named entities that the system is able to recognize. *Precision* measures the accuracy, i.e. the percentage of the labeled named entities that agree with the gold standard.

A third measure ($F_1$) is used to combine these two metrics as shown in the following:

$$Precision = \frac{C}{L} \qquad Recall = \frac{C}{G} \qquad F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Where:

- $L$: Number of labeled named entities
- $G$: Number of gold-standard named entities
- $C$: Number of correctly labeled named entities

The $F_1$ measure has been the de facto evaluation and optimization metric for named entity recognition, because of its simplicity and generality. However, there have been debates about how informative this metric really is. In a NLP blog

note,[6] Chris Manning compares various types of errors in NER and argues that $F_1$ penalizes some types of errors too much. For example, a perfect boundary recognition with incorrect categorization receives the same penalty as a total miss of a named entity. Furthermore, Manning shows that optimization for such an evaluation metric biases the system towards labeling fewer named entities.

### *7.2.7 Evaluation Campaigns*

Since its introduction, named entity recognition has been a popular subject for group evaluation. There have been three major NER evaluation campaigns as part of NLP conferences. The shared task at the 6th and the 7th Message Understanding Conference (MUC) were the first NER system competitions[7] which consisted of extracting entities like person, location, organization, temporal and number expressions [66]. The evaluation followed the template-filling framework of Information Extraction (IE) with the standard precision, recall metrics. MUC's evaluation counts partial credits for cases in which the boundary of the entity or its class are incorrect.

In 2002 and 2003, the Conference of Natural Language Learning (CoNLL) included a language-independent shared task on named entity recognition. These were important forums for language-independent NER[8] where a diverse set of learning techniques and features were explored. The BIO encoding of the NER problem, the addition of the miscellaneous (MISC) class of named entities[9] and also the exact matching criteria in the evaluations were protocols which were introduced in the CoNLL shared tasks and since then have been followed by many researchers.

The Automatic Content Extraction (ACE) program was a multilingual (Arabic, Chinese and English) program that was focused on tasks such as named entity recognition and mention detection [23]. The program has created substantial amount of gold-standard data for the three languages. The Arabic corpus is probably one the most important dataset for Semitic NER. ACE introduced a few new conventions for named entity recognition; in addition to the standard person, location and organization classes, ACE added additional entity types such as *facility*, *vehicle*, *weapon* and *geographic point entity (GPE)*. Furthermore, ACE used a more comprehensive evaluation framework. The evaluation incorporated several kinds of errors into an integrated scoring mechanism. This was aimed to address some of the concerns regarding the complete matching criteria of CoNLL.

---

[6] http://nlpers.blogspot.com/2006/08/doing-named-entity-recognition-dont.html

[7] The term *named entity* was first introduced at the MUC-6 [54].

[8] The 2002 shared task was conducted on Dutch and Spanish [67]. The 2003 shared task was conducted on English and German [68].

[9] Per CoNLL definition, any named entities that does not belong to the person, location and organization classes is considered to be MIS.

### 7.2.8 *Beyond Traditional Named Entity Recognition*

In the past decade, the scope of named entity recognition has been extended to new categories and topics. Depending on the topic, there can be various categories of named entities. Works such as [63] constructed extended ontology of named entity categories. These ontologies are useful for NER in multi-topic texts like Wikipedia or weblogs. Balasuriya et al. [8] highlight the substantial difference between entities appearing in English Wikipedia versus traditional corpora, and the effects of this difference on NER performance. There is evidence that models trained on Wikipedia data generally perform well on corpora with narrower domains. Nothman et al. [56] and Balasuriya et al. [8] show that NER models trained on both automatically and manually annotated Wikipedia corpora perform reasonably well on news corpora. The reverse scenario does not hold true for models trained on news text and there is a major performance drop.

It is no surprise that the state-of-the-art news-based NER systems perform less impressively when subjected to new topics and domains. Domain and topic diversity of named entities has been studied within the framework of domain adaptation research. In domain adaptation studies, the traditional domain which usually matches the labeled training data in most part is the *source* domain and the novel domain which usually lacks large amount of labeled data is the *target* domain. A group of these methods use semi-supervised learning frameworks such as self-training and select the most informative features and training instances to adapt a source domain learner to a new target domain. Wu et al. [71] bootstrap the NER learner with a subset of unlabeled instances that bridge the source and target domains. Jiang and Zhai [36] as well as [21] make use of some labeled target-domain data, augmenting the feature space of the source model with features specific to the target domain.

There is also a body of work on extraction of named entities from biological and medical text.[10] In these works, target named entities range from the names of enzymes and proteins in biology texts to symptoms, medicines and diseases in medical records.

## 7.3 Named Entity Recognition for Semitic Languages

Named entity recognition inherits many of the general problems of Semitic NLP; complex morphology, the optional nature of short vowels (diacritics) and generally the non-standard orthography are well known problems involved in the processing of Semitic languages which also affect NER.

Except Arabic, NER is an under-studied problem for other Semitic languages. There is small to medium amount of labeled data for Arabic and Hebrew NER

[10] See [43] and [45] for an overview Biomedical NER.

**Table 7.3** Examples of morphological and orthographic challenges in Semitic NER

| | Morphology | Orthography |
|---|---|---|
| Arabic | للأمريكيين (ل + ال + أمريكي + ين) | براد (برّاد/براد) |
| | llAmrykyyn (l + Al + Amryky + yn) | brAd (brrAd / brAd) |
| | *to the Americans* | *refrigerator / Brad* |
| Hebrew | באמריקא (ב + אמריקא) | אלון (אלוּן / אלון) |
| | bamrika (b + amrika) | alwn (alun / alon) |
| | *in America* | *to lodge / Alon* |

and for the rest of Semitic languages there is almost no resource. In the following sections we review the common challenges and some solutions for Semitic NER with a special focus on Arabic and Hebrew.

### *7.3.1 Challenges in Semitic Named Entity Recognition*

There are four main problems involved with Semitic languages which make Semitic NER a challenging task. Table 7.3 illustrates samples for some of these problems in Arabic and Hebrew.[11]

*Absence of capitalization:* For English and other Latin-scripted languages, the use of capitalization is a helpful indicator for named entities.[12] Maltese is the only Semitic language that uses capitalization in this similar fashion. The lack of capitalization in other Semitic languages like Arabic and Hebrew increases the ambiguity both in recognition and categorization of the named entities.

*Optional vowels*: Vowels are present in different levels in Semitic languages. Short vowels (diacritics) are optional in Arabic and Hebrew. In Amharic writing, vowels are mostly present (except in the case of gemination) and Maltese's Latin scripting explicitly incorporates vowels. Whenever vowels become optional (as they are in Hebrew and Arabic), ambiguity increases. For example in Table 7.3, the non-vocalized surface form of the Hebrew word *alwn* in can be interpreted as the verb *alun* or the person name *Alon*. Similarly, the Arabic token *BrAd* might refer to the Arabic noun *brrAd* (with an optional gemination) or the Western name *Brad*.

*Complex morphology*: The concatenative morphology in Semitic languages makes it possible for a named entity to get attached to different clitics and form a longer phrase. For example in Table 7.3, the Arabic entity (Amryky: American) is agglutinated to a the *Al* (definite) proclitic and the *yn* (plural) suffix and forms a noun phrase (the Americans). In order to recognize and categorize such entities,

[11] Samples for the Arabic are shown using the Buckwalter romanization [18] and samples for the Hebrews are shown using the romanization scheme in [40].

[12] Capitalization is not used consistently among Latin-scripted languages. Capitalization typically applies to proper nouns in English, to all nouns in German, and to any important noun in Italian.

morphological analysis needs to be performed. Thus, morphological analysis and disambiguation is expected to play an important role in Semitic NER.

*Transliteration and diversity of spelling:* Multiple transliteration of named entities is a common problem in most languages including the Semitic family. The non-standard mapping of cross-lingual consonants results in various spellings of phonologically complex names such as *Schwarzenegger* in Arabic or Hebrew. Moreover, in most Semitic languages we observe some diversity of spelling both for local and foreign names. For example, the first letter of person name Haylü in Amharic can take multiple forms which results in six different spellings of the name [65]. Another example is the multiple mapping between the "h" or "t" consonants in the Roman languages to Arabic.[13]

### *7.3.2 Approaches to Semitic Named Entity Recognition*

There is an extensive body of works on Arabic named entity recognition. That includes the creation of gazetteers, labeled datasets, statistical and also rule-based systems. The system in [64] is an example of a rule-based approach. The approach includes creation of name lists for the named entities and non-entities (white and black lists) along with the extraction rules (in form of regular expressions). The RENAR system [73] is a more recent rule-based approach. It is based on searching gazetteers followed by a set of hand-crafted grammar recognition rules for extracting out of lexicon entities. Finally, the system of [57] is a more recent hybrid approach in combining a rule-based system with various statistical classifiers in extracting a large set of named entity classes.

A range of statistical learning algorithms have been applied to Arabic NER: Nezda et al. [55] and Benajiba et al. [11] use Maximum Entropy, Benajiba et al. [12], Abdul-Hamid and Darwish [1] use Support Vector Machines and Farber et al. [24] as well as [53] use Perceptron. A range of lexical, morphological and syntactic features have been used in these statistical systems. The development and the distribution of tools such as MADA [30] and AMIRA [22] and SAMA [46] led to studies on the role and effects of morphological features in Arabic named entity recognition. Moreover, the English translation information provided by MADA has provided useful bilingual features. For example, Farber et al. [24] use the gloss translations to estimate a capitalization feature for Arabic words. In other studies such as [12], the MADA package has been extensively used to explore different morphological features with different learning frameworks. In the next section we will review the work in [12] as a case study for Semitic NER.[14]

[13] For example, foreign person names such as Hayato (Japanese) or Tahvo (Finish) can be mapped to different Arabic spellings.

[14] Other relevant works on Arabic NER: [25, 47, 50, 62].

There are two major published works on Hebrew NER. Lemberski [44][15] uses a Maximum Entropy sequence classifier and a set of lexical and morphological features. Features include lexeme, POS tag, several named entity lexicon and information extracted from hand-crafted regular expression patterns. In order to train the system with labeled data, a morphologically tagged corpus was manually annotated with the named entity information. The annotation was in the framework of MUC-7 on a set of 50 Hebrew news articles. In an extended work, Ben Mordecai and Elhadad [14] use three systems separately and jointly for Hebrew named entity recognition. In the following section we will review this work as a case study for Semitic NER.

Similar to English, the majority of the systems for Arabic and Hebrew NER are trained and evaluated on the news corpora. The named entity categories usually include the traditional person, organization, location classes. Some of the Arabic NER works go beyond the traditional classes and introduce additional classes relevant to the domain. Shaalan and Raza [64] extract ten named entity classes related to the business news domain. Some of the numeric classes are non-conventional (e.g. phone number) and contributed to the development of new labeled dataset for evaluation. The system in [55] uses an extensive annotation of text from the Arabic Tree Bank with 18 classes of named entities. The categories include several quantitative and temporal classes such as money and time.

Arabic Wikipedia has been the test-bed for a few recent studies on named entity recognition. Mohit et al. [53] demonstrate that traditional named entity classes are insufficient for a multi-topic corpus like Wikipedia. They use a relaxed annotation framework in which article-specific classes are considered and labeled. For example, for an article about *Atom*, annotators introduced and labeled particle names (e.g. electron, proton). Furthermore, Mohit et al. [53] develop an NER system which recognize (but does not categorize) their extended set of named entity classes for Arabic Wikipedia. Extended classes of named entities have also been used as a taxonomy for Arabic Wikipedia. Alotaibi and Lee [4] use a supervised classification framework to assign Wikipedia articles to one of their eight coarse-grained named entity classes.

Semitic NER has been studied as part of other relevant tasks. For example, Kirschenbaum and Wintner [40] locate named entities for the purpose of translating them from Hebrew to English. We will review these works in Sect. 7.5 along with other works relevant to Semitic NER.

## 7.4 Case Studies

In this section we review the work of Benajiba et al. [12] and also Ben Mordecai and Elhadad [14] as case studies in (respectively) Arabic and Hebrew named entity recognition. The two works share a common approach to Semitic NER: Exploring

[15] Published in Hebrew.

different learning algorithms and features sets and also lexicon construction to achieve an optimal performance. Benajiba et al. [12] aim at finding the optimal feature set for different classes of Arabic named entities. Ben Mordecai and Elhadad [14] include a brief analysis of effective features, but mainly focus on combining different learning methods for optimizing Hebrew NER. In the following we review different aspects of these two works:

### *7.4.1 Learning Algorithms*

The system in [12] is an empirical framework to study the effects of different features on Arabic NER. It uses two discriminative learners (support vector machines and conditional random fields) to construct classifiers for each named entity class. Thus, there are classifiers for the person class, location class, etc. that label the named entity boundaries. After the initial per-class labeling, a collective NER classification takes place with a voting mechanism.

Ben Mordecai and Elhadad [14] explore a baseline rule-based system made of regular expressions and two statistical classifiers (Hidden Markov Model and Maximum Entropy). After trying different HMM schemes, they chose a structure where each state is made of a named entity class joined with the POS tag. Moreover, the HMM states omit a feature representation of the words. By such joint inclusion of the class label and the POS tag, they incorporate some structural knowledge in to their model. In contrast, their standard maximum entropy model of NER is not constrained and freely uses features independent of each other.

### *7.4.2 Features*

Feature selection is an important component of these two case studies and also most other Arabic and Hebrew NER studies. As discussed earlier, NER is a heavily lexicalized task and models rely strongly on lexical and contextual features. A standard set of contextual features such as the preceding and following tokens and morphemes are inherited from the English systems. Furthermore, morphological complexities of Semitic languages requires explicit inclusion of morphological features into the models. In Arabic, for example the gender or number agreements between adjacent proper nouns are important hints to find the spans of the named entity. In the absence of robust morphological and syntactic analyzers (e.g. in Hebrew systems), models benefit from shallow structural and morphological features such as affixes or the token's position in the sentence.

Table 7.4 compares features used in our two cases studies [12,14]. The feature set used in the Arabic system includes lexical, contextual features and morphological features as well as features from named entity lexicons built from resources like Wikipedia. Most of the morphological features are extracted by using the Arabic MADA toolkit. The effectiveness of features has been estimated for each of the

**Table 7.4** Features in the Arabic [12] and the Hebrew [14] systems

| Feature | Arabic | Hebrew |
|---|---|---|
| Context (prev. and follow. $n$ tokens) | × | × |
| Affixes (shallow morphology) tag | × | × |
| POS tag | × | × |
| Gazeteers | × | × |
| Base phrase chunk | × | |
| Corresponding English capitalization | × | |
| Morphological analysis (person, gender, number, etc.) | × | |
| Frequency features (being a frequent nouns, phrase, token) | | × |
| Structural features (token's position in the sentence) | | × |
| Regular expressions | | × |
| Lemma | | × |

named entity classes. Some of these features tend to be contributing for most named entity classes (e.g. the morphological aspect or English capitalization). However, because each class holds its own classifier and feature analysis, there is not always a strong consensus about the general effectiveness of a certain feature.

The feature set in [14] comprises of morphological, structural lexical and contextual features. For morphological features there is not much Hebrew-specific analysis and they are limited to POS tags, affixes and the lemma. However, there is a set of regular expressions and structural features which provide some language specific flavor to the model. Furthermore, gazetteer features use a few lexicons that hold a comprehensive list of frequent nouns and expressions and also use geographical and organizational lists.

### *7.4.3 Experiments*

Both studies use system combination algorithms. However, the combination is aimed toward different goals. For Benajiba et al. [12], each entity class has a separate classifier and feature set. The feature-based ranking framework (Fuzzy Borda Voting Scheme) is a mechanism to combine these different classifiers into one final classifier. There is an average of 2 % improvement in the $F_1$ score after reaching the optimum feature set of classifier voting. The support vector machines classifier outperforms others for the majority of classes and datasets while lexical features are the most contributing ones in most experiments.

System combination in [14] is based on a simple recall-oriented heuristic: Take the output of the best individual system (maximum entropy) and use the other two taggers as the back-off. Finally, the empirical experiments show that dictionary features along with the POS tag tend to be the most contributing features.

To summarize, the Arabic system in [12] and the Hebrew system in [14] are successful examples of Semitic NER using a hybrid mixture of supervised learners.

Both systems explore language-specific aspects of the problem, but in different ways; Ben Mordecai and Elhadad [14] use language-specific regular expressions to locate potential entities. Benajiba et al. [12] explicitly incorporate linguistic knowledge (e.g. Arabic morphology) as features in to its hybrid learning framework.

## 7.5 Relevant Problems

The importance of named entities for multilingual applications such as machine translation and cross language information retrieval has led researchers to focus on a few other problems which overlap with NER. Here we have a brief overview on three of such problems where Semitic languages (Arabic and Hebrew) have been studied.

### *7.5.1 Named Entity Translation and Transliteration*

The multilingual named entity information is useful for applications such as cross language information retrieval or machine translation. For example, Hermjakob et al. [32] have shown that inclusion of transliteration information improves machine translation quality. Also, Babych and Hartley [7] showed that incorporation of bilingual named entity information in general improves machine translation quality.

Named entities usually are either translated or transliterated across languages. Compound named entities which are composed of simple nominals (as opposed to proper nouns) might be translated across languages. For example an organizational entity like *The State Department* usually gets translated. In contrast, named entities composed of proper nouns such as *IBM* or *Adidas* usually get transliterated across languages. Table 7.5 presents examples of translation and transliterations for Arabic and Hebrew named entities.

There is a body of work on translation and transliteration of named entities for Arabic and Hebrew. Al-Onaizan and Knight[3] address the named entity translation problem. Their approach has two folds: baseline translation and transliteration of the named entities and later, a filtering based on the target language corpus. The underlying assumption is based on the occurrences of the named entities in the international news: names which are important and frequent in the source language (Arabic), are also frequent in the target language (English).

An important decision for a multilingual system (e.g. machine translation) is whether to translate or transliterate a given source language named entity. Hermjakob et al. [32] address this problem using a supervised classification approach. They use a parallel corpus of phrases which include bilingual transliterated name pairs. The Arabic side of the transliterated bitext is used to train a classifier which highlights words of a monolingual (Arabic) text that can be

**Table 7.5** Translation vs. transliteration of named entities in Arabic and Hebrew

| | | Translation | Transliteration |
|---|---|---|---|
| Arabic | **Source:** | AlbHr AlmtwsT / البحر المتوسط | rwmAnsyp / رومنسية |
| | **Gloss:** | the-sea the-middle | Romantisism |
| | **Translation:** | *Mediterranean Sea* | *Romantisism* |
| Hebrew | **Source:** | hayam hatichon / הים התיכון | eqzistentzializm / אקזיסטנציאליזם |
| | **Gloss:** | the-sea the-central | Existentialism |
| | **Translation:** | *Mediterranean Sea* | *Existentialism* |

transliterated. Similar classification frameworks have also been examined for the decision making of translation vs. transliteration for Hebrew[28, 40].

Machine transliteration deals with named entities that are translated with *preserved pronunciation* [38]. There are specific challenges in Arabic and Hebrew orthography and phonetics which add to the transliteration challenge. These include the optional nature of vowels, the absence of certain sounds (e.g. *p* in Arabic), zero or many mapping of certain sounds to Latin-based letters (e.g. multiple *h* in Arabic or *khaf* in Hebrew). An earlier approach to the problem is described in [2] which is a hybrid combination of phonetic-based and spelling-based models. The extracted transliterations are post-processed by a target language (English) spell checker. There are also transliteration studies which do not involve transliterating the term from scratch. In [32], the transliterated candidates are extracted from a bilingual phrase corpus and the transliteration problem is practically converted to a search problem. There, the system uses a scoring function to filter out the noisy transliterations using a large English corpus.[16] In a relevant framework, the work of Azab et al. [6] aims at automating the English to Arabic translation vs. transliteration decision and reducing the out of vocabulary terms of the MT system. They model the decision as a binary classification problem and later use their classifier within a SMT pipeline to direct a subset of source language named entities to a transliteration module.

For Hebrew, Goldberg and Elhadad [28] identify the borrowed and transliterated words. Their decision is binary: A word is either generated by a Hebrew language model, or by a foreign language model. They train a generative classifier using a noisy list of borrowed words along with regular Hebrew text. The work of Kirschenbaum and Wintner in [40] is also an effort to locate and transliterate the appropriate Hebrew terms. The framework is a single-class classifier which locates entities that are supposed to be transliterated.

[16]For more information about Arabic transliteration, see: [29].

**Table 7.6** An Arabic example of entity detection and tracking with gloss and literal translations

| | |
|---|---|
| Source | **أوباما** قام بحلف اليمين بوصفه عضو مجلس الشيوخ **وهو** خامس عضو بمجلس الشيوخ من أصول إفريقية.<br>**AwbAmA** qAm bHlf Alymyn bwSfh EDw mjls Al$ywx **whw** xAms EDw bmjls Al$ywx mn >Swl >fryqyp |
| Gloss | **AwbAmA:*Obama*** qAm bHlf:*sworn* Alymyn:*right* bwSfh:*as* EDw:*member* mjls:*parliment* Al$ywx:*the-experts* **whw:*and-he*** xAms:*the-fifth* EDw:*member* bmjls:in-parliment Al$ywx:the-experts mn:from >Swl:descent >fryqyp:African |
| Translation | *Obama took oath as a senate member which is the fifth African-American senator* |

### *7.5.2 Entity Detection and Tracking*

Mention detection is a subtask of information extraction which is focused on the identification of entities and the tracking of their associations to each other. Mentions can be named entities, nominals, or pro-nominals. Table 7.6 presents an Arabic example of entity detection, along with gloss and literal translations. A detection system is expected to highlight and link the two bold segments of the Arabic example. Entity detection is usually modeled as a sequence classification task where each token in a sentence gets assigned to an entity within the sentence. Similar to NER, there are tokens which are independent of entities and get an *O* label. The detection part of the task is similar to the NER. The tracking part might involve a separate linking model and coreference decoding.

Arabic mention detection was one of the tasks introduced in the ACE program. Florian et al. [27] presented a multi-lingual system which included an Arabic mention detection component. Their system uses two Maximum Entropy models, one for the detection and the other one for tracking. The tracking component is a binary linking model where each token gets either linked to another entity or starts a new entity. Also, there have been two recent studies on the effects of morphology and syntactic analysis on Arabic mention detection[9, 10] in which, richer Arabic linguistic knowledge boosted the performance.

### *7.5.3 Projection*

Availability of parallel corpora, automatic word alignment and translation systems resulted in a body of work on resource projection [72]. In a projection framework we use a word-aligned corpus to *project* some linguistic information (e.g. named entity boundaries) from a language (e.g. English) to another language (e.g. Hebrew). This has been a useful framework for equipping resource-poor languages with some labeled data. Projection is not always a deterministic operation and cross lingual

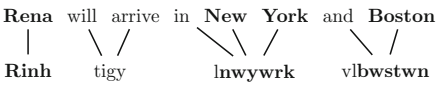


**Fig. 7.3** An NER projection example from English to Hebrew

differences can make it a challenging task. Figure 7.3 demonstrate an example of named entity projection from English to Hebrew. It can be seen that morphological richness of the Hebrew does not allow a 1-1 entity mapping across two languages. Thus morphological analysis and segmentation should be considered as part of the a projection pipeline.

There have been some successful attempts on the projection of entity information for Arabic. Hassan et al. [31] extract bilingual named entity pairs from parallel and comparable corpora using similarity metrics that use phonetic and translation model information. Zitouni and Florian [74] study the use of projection (through English to Arabic machine translation) to improve Arabic mention detection. Benajiba and Zitouni [10] directly project the mention detection information using automatic word alignments. The projected Arabic corpus provides new features which augments and improves the baseline Arabic mention detection system. Huang et al. [34] study the problem of finding various English spelling of Arabic names which affects machine translation and information extraction systems. They use a projection framework to locate various spelling of a given Arabic name.

## 7.6 Labeled Named Entity Recognition Corpora

Similar to the research, data resources for the Semitic NER have been limited to Arabic and Hebrew. The Automatic Content Extraction (ACE) program is a multilingual information extraction effort focused on Arabic, Chinese and English. Over the past decade, Arabic has been one of the focus languages of the Entity Detection and Tracking (EDT) task of the ACE. As a result, ACE has prepared a few standard Arabic corpora with named entity information [70]. These corpora are primarily in the newswire domain with recent additions of weblogs and broadcast news text. The named entity categories are targeted towards the political news. They include Person, Location, Organization, Facility, Weapon, Vehicle and Geo-Political Entity (GPE). The Arabic named entity annotations are performed with character-level information which boosts the accuracy of the data for morphologically compound tokens.[17] ACE has been releasing most of its dataset through the Linguistic Data Consortium (LDC).

In addition to the standard ACE datasets, a few projects have resulted in annotation of new NER datasets. The Ontonotes project [33] is an ongoing large scale multilingual annotation effort with several layers of linguistic information on texts collected from a diverse set of domains. Arabic Ontonotes includes annotation

[17] See [42] for more information about the Arabic ACE dataset.

of parsing, word senses, coreferences and named entities.[18] The publicly released[19] Arabic ANER corpus [11] is a token-level annotated newswire corpora with four named entity classes: person, location, organization and miscellaneous. Mohit et al. [53] also have released a corpus of Arabic Wikipedia articles with an extended set of named entity categories. Finally, Attia et al. [5] created a large scale lexicon of Arabic named entities from resources such as Wikipedia.[20]

Named entity annotation for Hebrew has been limited to a few projects that we discussed earlier. Hebrew corpus annotation of named entities are reported in [14, 44]. Furthermore, the annotated corpora in [35] includes a layer of named entity information.

## 7.7 Future Challenges and Opportunities

Named entity recognition is still far from a solved problem for Semitic languages. Amharic, Syriac and Maltese lack the basic data resources for building a system. The $F_1$ performance of the best Arabic and Hebrew systems varies between 60 and 80 % depending on the text genres. Most of the available labeled datasets are mainly news wire corpora which might degrade the NER performance in other topics and domains.

There are many interesting open questions to be explored. For the low resource languages like Amharic or Syriac, well established frameworks such as active learning or projection can be explored to create the basic data requirements and estimating basic models. Online resources such as Wikipedia can also provide the basic named entity corpora and lexicons.[21]

For medium-resource languages like Arabic and Hebrew, NER needs to be tested in new topics and genres with extended named entity classes. To do so, semi-supervised learning frameworks along with domain adaptation methods are the natural starting solutions. Morphological information plays an important role in Semitic NER. Thus, richer incorporation of morphology in NER models in form of joint modeling is an interesting avenue to explore. Moreover richer linguistic information such as constituency and dependency parsing, semantic resources such as the Wordnet and Ontonotes are expected to enrich NER models.

[18] The fourth release of Ontonotes includes named entity annotation for a corpus of 300,000 words.

[19] currently at http://www1.ccls.columbia.edu/~ybenajiba/downloads.html

[20] Work presented in [55, 64] also report a large scale annotation of named entity information. However the datasets were not released publicly.

[21] According to Wikipedia statistics, Amharic Wikipedia has more than 10,000 articles which is a promising resource for gazetteer construction.

## 7.8 Summary

We reviewed named entity recognition (NER) as an important task for processing Semitic languages. We first sketched an overview of NER research, its history and the current state of the art. We followed with problems specific to Semitic NER and reviewed a wide range of approaches for Arabic and Hebrew NER. We observed that complex morphology and the lack of capitalization create additional challenges for Semitic NER. We focused on two case studies for Arabic and Hebrew and reviewed their learning frameworks and features. Moreover, we explored the state of data resources and research on relevant tasks such as named entity translation, transliteration and projection for Hebrew and Arabic. We concluded that Semitic NER is still an open problem. For low resource languages such as Amharic and Syriac basic data resources are still needed for constructing baseline systems. For Arabic and Hebrew, inclusion of richer linguistic information (e.g. dependency parsing) and adaptation of the current systems to new text domains are interesting avenues to explore.

**Acknowledgements** I am grateful to Kemal Oflazer, Houda Bouamor, Emily Alp and two anonymous reviewers for their comments and feedback. This publication was made possible by grant YSREP-1-018-1-004 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

1. Abdul-Hamid A, Darwish K (2010) Simplified feature set for Arabic named entity recognition. In: Proceedings of the 2010 named entities workshop. Association for Computational Linguistics, Uppsala, pp 110–115
2. Al-Onaizan Y, Knight K (2002a) Machine transliteration of names in Arabic texts. In: Proceedings of the ACL-02 workshop on computational approaches to Semitic languages, Philadelphia. Association for Computational Linguistics
3. Al-Onaizan Y, Knight K (2002b) Translating named entities using monolingual and bilingual resources. In: Proceedings of 40th annual meeting of the Association for Computational Linguistics, Philadelphia. Association for Computational Linguistics, pp 400–408
4. Alotaibi F, Lee M (2012) Mapping Arabic Wikipedia into the named entities taxonomy. In: Proceedings of COLING 2012: posters, Mumbai. The COLING 2012 Organizing Committee, pp 43–52
5. Attia M, Toral A, Tounsi L, Monachini M, van Genabith J (2010) An automatically built named entity lexicon for Arabic. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Rosner M, Tapias D (eds) Proceedings of the seventh conference on international language resources and evaluation (LREC'10), Valletta. European Language Resources Association (ELRA)
6. Azab M, Bouamor H, Mohit B, Oflazer K (2013) Dudley North visits North London: learning when to transliterate to arabic. In: Proceedings of the 2013 conference of the North American chapter of the Association for Computational Linguistics: human language technologies (NAACL-HLT 2013), Atlanta. Association for Computational Linguistics
7. Babych B, Hartley A (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, EAMT '03, Dublin

8. Balasuriya D, Ringland N, Nothman J, Murphy T, Curran JR (2009) Named entity recognition in Wikipedia. In: Proceedings of the 2009 workshop on the people's web meets NLP: collaboratively constructed Semantic resources. Association for Computational Linguistics, Suntec, pp 10–18
9. Benajiba Y, Zitouni I (2009) Morphology-based segmentation combination for Arabic mention detection. ACM Trans Asian Lang Inf Process (TALIP) 8:16:1–16:18
10. Benajiba Y, Zitouni I (2010) Enhancing mention detection using projection via aligned corpora. In: Proceedings of the 2010 conference on empirical methods in natural language processing, Cambridge. Association for Computational Linguistics, pp 993–1001
11. Benajiba Y, Rosso P, BenedíRuiz JM (2007) ANERsys: an Arabic named entity recognition system based on maximum entropy. In: Gelbukh A (ed) Proceedings of CICLing, Mexico City. Springer, pp 143–153
12. Benajiba Y, Diab M, Rosso P (2008) Arabic named entity recognition using optimized feature sets. In: Proceedings of the 2008 conference on empirical methods in natural language processing, Honolulu. Association for Computational Linguistics, pp 284–293
13. Bender O, Och FJ, Ney H (2003) Maximum entropy models for named entity recognition. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, Edmonton, pp 148–151
14. Ben Mordecai N, Elhadad M (2005) Hebrew named entity recognition. Master's thesis, Department of Computer Science, Ben Gurion University of the Negev
15. Berger AL, Pietra VJD, Pietra SAD (1996) A maximum entropy approach to natural language processing. Comput Linguist 22:39–71
16. Bikel D, Miller S, Schwarz R, Weischedel R (1997) Nymble: a high-performance learning name-finder. In: Proceedings of the applied natural language processing, Tzigov Chark
17. Borthwick A (1999) A maximum entropy approach to named entity recognition. Phd thesis, Computer Science Department, New York University
18. Buckwalter T (2002) Buckwalter Arabic morphological analyzer version 1.0
19. Chieu HL, Ng HT (2002) Named entity recognition: a maximum entropy approach using global information. In: Proceedings of the 19th international conference on computational linguistics – vol 1, COLING '02. Taipei
20. Collins M (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 (EMNLP), Philadelphia, pp 1–8
21. Daumé III H (2007) Frustratingly easy domain adaptation. In: Proceedings of the 45th annual meeting of the Association of Computational Linguistics, Prague. Association for Computational Linguistics, pp 256–263
22. Diab M (2009) Second generation tools (AMIRA 2.0): fast and robust tokenization, pos tagging, and base phrase chunking. In: Proceedings of the 2nd international conference on Arabic language resources and tools, Cairo
23. Doddington G, Mitchell A, Przybocki M, Rambow L, Strassel S, Weischedel R (2004) The automatic content extraction (ACE) program-tasks, data and evaluation. In: Proceedings of LREC 2004, Lisbon, pp 837–840
24. Farber B, Freitag D, Habash N, Rambow O (2008) Improving NER in Arabic using a morphological tagger. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odjik J, Piperidis S, Tapias D (eds) Proceedings of the sixth international language resources and evaluation (LREC'08), Marrakech. European Language Resources Association (ELRA), Marrakesch, pp 2509–2514
25. Fehri H, Haddar K, Ben Hamadou A (2011) Recognition and translation of Arabic named entities with NooJ using a new representation model. In: Proceedings of the 9th international workshop on finite state methods and natural language processing, Blois. Association for Computational Linguistics, pp 134–142
26. Florian R, Ittycheriah A, Jing H, Zhang T (2003) Named entity recognition through classifier combination. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, Edmonton, pp 168–171

27. Florian R, Hassan H, Ittycheriah A, Jing H, Kambhatla N, Luo X, Nicolov N, Roukos S (2004) A statistical model for multilingual entity detection and tracking. In: Dumais S, Marcu D, Roukos S (eds) Proceedings of the human language technology conference of the North American chapter of the Association for Computational Linguistics: HLT-NAACL 2004, Boston. Association for Computational Linguistics
28. Goldberg Y, Elhadad M (2008) Identification of transliterated foreign words in Hebrew script. In: Proceedings of the 9th international conference on Computational linguistics and intelligent text processing, CICLing'08, Haifa, pp 466–477
29. Habash N, Soudi A, Buckwalter T (2007) On arabic transliteration. Text Speech Lang Technology 38:15–22
30. Habash N, Rambow O, Roth R (2009) MADA+TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In: Choukri K, Maegaard B (eds) Proceedings of the second international conference on Arabic language resources and tools, the MEDAR consortium, Cairo
31. Hassan A, Fahmy H, Hassan H (2007) Improving named entity translation by exploiting comparable and parallel corpora. In: Proceedings of the conference on recent advances in natural language processing (RANLP '07), Borovets
32. Hermjakob U, Knight K, Daumé III H (2008) Name translation in statistical machine translation – learning when to transliterate. In: Proceedings of ACL-08: HLT, Columbus. Association for Computational Linguistics, pp 389–397
33. Hovy E, Marcus M, Palmer M, Ramshaw L, Weischedel R (2006) OntoNotes: the 90% solution. In: Proceedings of the human language technology conference of the NAACL (HLT-NAACL), New York City. Association for Computational Linguistics, pp 57–60
34. Huang F, Emami A, Zitouni I (2008) When Harry met Harri: cross-lingual name spelling normalization. In: Proceedings of the 2008 conference on empirical methods in natural language processing, Honolulu. Association for Computational Linguistics, pp 391–399
35. Itai A, Wintner S (2008) Language resources for Hebrew. Lang Resour Eval 42:75–98
36. Jiang J, Zhai C (2006) Exploiting domain structure for named entity recognition. In: Proceedings of the human language technology conference of the NAACL (HLT-NAACL), New York City. Association for Computational Linguistics, pp 74–81
37. Jurafsky D, Martin JH (2008) Speech and language processing. Pearson Prentice Hall, Upper Saddle River
38. Karimi S, Scholer F, Turpin A (2011) Machine transliteration survey. ACM Comput Surv 43:17:1–17:46
39. Khalid MA, Jijkoun V, De Rijke M (2008) The impact of named entity normalization on information retrieval for question answering. In: Proceedings of the IR research, 30th European conference on advances in information retrieval, Glasgow, Springer, pp 705–710
40. Kirschenbaum A, Wintner S (2009) Lightly supervised transliteration for machine translation. In: Proceedings of the 12th conference of the European chapter of the ACL (EACL 2009), Athens. Association for Computational Linguistics, pp 433–441
41. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, ICML '01, Williamstown. Morgan Kaufmann, pp 282–289
42. LDC (2005) ACE (automatic content extraction) Arabic annotation guidelines for entities, version 5.3.3. Linguistic Data Consortium, Philadelphia
43. Leaman R, Gonzalez G (2008) Banner: an executable survey of advances in biomedical named entity recognition. In: Proceedings of pacific symposium on biocomputing, Kohala Coast, pp 652–663
44. Lemberski G (2003) Named entity recognition in Hebrew. Master's thesis, Department of Computer Science, Ben Gurion University
45. Leser U, Hakenberg J (2005) What makes a gene name? Named entity recognition in the biomedical literature. Brief. Bioinform 6:357–369
46. Maamouri M, Graff D, Bouziri B, Krouna S, Bies A, Kulick S (2010) LDC standard Arabic morphological analyzer (SAMA) version 3.1, LDC2004L02. Linguistic Data Consortium, Philadelphia

47. Maloney J, Niv M (1998) TAGARAB: a fast, accurate arabic name recognizer using high precision morphological analysis. In: Proceedings of the workshop on computational approaches to Semitic languages, Montreal
48. Malouf R (2002) Markov models for language-independent named entity recognition. In: Proceedings of the 6th conference on natural language learning – vol 20, COLING-02, Stroudsburg. Association for Computational Linguistics, pp 1–4
49. McCallum A, Li W (2003) Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Daelemans W, Osborne M (eds) Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, Edmonton, pp 188–191
50. Mesfar S (2007) Named entity recognition for Arabic using syntactic grammars. In: Kedad Z, Lammari N, Métais E, Meziane F, Rezgui Y (eds) Natural language processing and information systems. Lecture notes in computer science, vol 4592. Springer, Berlin, pp 305–316
51. Mikheev A, Moens M, Grover C (1999) Named entity recognition without gazetteers. In: Proceedings of the ninth conference of the European chapter of the Association for Computational Linguistics (EACL-99), Bergen. Association for Computational Linguistics
52. Miller S, Crystal M, Fox H, Ramshaw L, Schwartz R, Stone R, Weischedel R, The Annotation Group (1998) Algorithms that learn to extract information BBN: description of the sift system as used for MUC-7. In: Proceedings of the seventh message understanding conference (MUC-7), Fairfax
53. Mohit B, Schneider N, Bhowmick R, Oflazer K, Smith NA (2012) Recall-oriented learning of named entities in arabic wikipedia. In: Proceedings of the 13th Conference of the European Chapter of the ACL (EACL 2012), Avignon. Association for Computational Linguistics
54. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes 30:3–26
55. Nezda L, Hickl A, Lehmann J, Fayyaz S (2006) What in the world is a *Shahab*? Wide coverage named entity recognition for Arabic. In: Proccedings of LREC, Genoa, pp 41–46
56. Nothman J, Murphy T, Curran JR (2009) Analysing Wikipedia and gold-standard corpora for NER training. In: Proceedings of the 12th conference of the European chapter of the ACL (EACL 2009), Athens. Association for Computational Linguistics, pp 612–620
57. Oudah M, Shaalan K (2012) A pipeline Arabic named entity recognition using a hybrid approach. In: Proceedings of COLING 2012, Mumbai. The COLING 2012 Organizing Committee, pp 2159–2176
58. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77:257–286
59. Ratinov L, Roth D (2009) Design challenges and misconceptions in named entity recognition. In: Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009), Colorado. Association for Computational Linguistics, Boulder, pp 147–155
60. Riloff E, Jones R (1999) Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the sixteenth national conference on artificial intelligence and the eleventh innovative applications of artificial intelligence conference innovative applications of artificial intelligence, Orlando. American Association for Artificial Intelligence, pp 474–479
61. Riloff EM, Phillips W (2004) Introduction to the sundance and autoslog systems. Technical report, University of Utah
62. Samy D, Moreno A, Guirao JM (2005) A proposal for an Arabic named entity tagger leveraging a parallel corpus. In: Proceedings of the conference of the recent advances in natural language processing (RANLP-05), Borovets
63. Sekine S, Sudo K, Nobata C (2002) Extended named entity hierarchy. In: Proceedings of LREC, Las Palmas
64. Shaalan K, Raza H (2009) NERA: named entity recognition for Arabic. J Am Soc Inf Sci Technol 60(8):1652–1663
65. Sintayehu Z (2001) Automatic classification of Amharic news items: the case of Ethiopian news agency. Master's thesis, School of Information Studies for Africa, Addis Ababa University

66. Sundheim BM (1995) Named entity task definition, version 2.1. In: Proceedings of the Sixth Message Understanding Conference (MUC-6), Columbia
67. Tjong Kim Sang EF (2002) Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In: Proceedings of the sixth conference on natural language learning (CoNLL-2002), Taipei
68. Tjong Kim Sang EF, De Meulder F (2003) Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Daelemans W, Osborne M (eds) Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, Edmonton, pp 142–147
69. Toral A, Noguera E, Llopis F, Muñoz R (2005) Improving question answering using named entity recognition. In: Natural language processing and information systems, vol 3513/2005. Springer, Berlin/New York, pp 181–191
70. Walker C, Strassel S, Medero J, Maeda K (2006) ACE 2005 multilingual training corpus. LDC2006T06, Linguistic Data Consortium, Philadelphia
71. Wu D, Lee WS, Ye N, Chieu HL (2009) Domain adaptive bootstrapping for named entity recognition. In: Proceedings of the 2009 conference on empirical methods in natural language processing, Singapore. Association for Computational Linguistics, pp 1523–1532
72. Yarowsky D, Ngai G, Wicentowski R (2001) Inducing multilingual text analysis tools via robust projection across aligned corpora. In: Proceedings of the first international conference on human language technology research, HLT '01, Stroudsburg. Association for Computational Linguistics, pp 1–8
73. Zaghouani W (2012) RENAR: A rule-based Arabic named entity recognition system. ACM Trans Asian Lang Inf Process (TALIP) 11:1–13
74. Zitouni I, Florian R (2008) Mention detection crossing the language barrier. In: Proceedings of the 2008 conference on empirical methods in natural language processing, Honolulu. Association for Computational Linguistics, pp 600–609