2020

Exploration of Approaches to Arabic Named Entity Recognition

Husamelddin Balla

Sarah Jane Delaney

Exploration of Approaches to Arabic Named Entity Recognition

Husamelddin A.M.N Balla and Sarah Jane Delany

Technological University Dublin
School of Computer Science
Dublin, Ireland
http://www.tudublin.ie
{husamelddin.balla,sarahjane.delany}@tudublin.ie

Abstract. The Named Entity Recognition (NER) task has attracted significant attention in Natural Language Processing (NLP) as it can enhance the performance of many NLP applications. In this paper, we compare English NER with Arabic NER in an experimental way to investigate the impact of using different classifiers and sets of features including language-independent and language-specific features. We explore the features and classifiers on five different datasets. We compare deep neural network architectures for NER with more traditional machine learning approaches to NER. We discover that most of the techniques and features used for English NER perform well on Arabic NER. Our results highlight the improvements achieved by using language-specific features in Arabic NER.

Keywords: Named Entity Recognition · Machine Learning · Arabic NER.

1 Introduction

Named Entity Recognition (NER) is the process of identifying the proper names in text and classifying them as one of a set of predefined categories of interest. There are three universally accepted categories which are the names of locations, people and organisations. There are other common categories such as recognition of time/date expressions, measures (money, percent, weight etc.), email addresses etc. In addition, there can be domain-specific categories such as the names of medical conditions, drugs, bibliographic references, names of ships, etc. NER is useful for applications such as question answering, information retrieval, information extraction, automatic summarization, machine translation and text mining [1].

Arabic is one of the five official languages used by the United Nations. Approximately 360 million people speak Arabic in more than 25 countries and

Arabic script represents 8.9% of the world's languages [2]. Although there is existing work on Arabic NER, it still in the primary stage compared with English NER [2]. Certain characteristics of the Arabic language offer challenges for the task of NER. Unlike English and other European languages, capitalization does not exist in Arabic script. Thus, employing capitalization as a feature in Arabic NER is not an option. However, translation to English is one way to solve this problem [3]. The Arabic language is morphologically complicated, a word may consist of prefixes, lemma and suffixes in different combinations [4]. That can affect the performance in Arabic NER as typically features derived from the suffix and affix of the words are used. Also, spelling alternates can be a challenge in Arabic NER. In the Arabic language, words (including named entities) may be spelt in different ways but have the same exact meaning generating a many-to-one ambiguity [2]. The lack of resources in Arabic presents another challenge for Arabic NER. There is a lack of the freely available Arabic datasets and gazetteers as many of the available ones are not appropriate for Arabic NER tasks because of the absence of NEs annotations.

In this paper we explore approaches for NER on Arabic text to determine how the state of the art approaches to NER work on the Arabic language. We investigate the impact of using different classifiers and sets of features including both language-independent and language-specific features, testing them on five different datasets. We have taken English as the second source language in our work because English NER is the most developed among other NER models. Recently, research on English NER have achieved the best performance in the field and represents the state of the art. We also compare against the more recent deep neural network approaches. The neural network approaches were found to perform better than the traditional machine learning approaches for both Arabic and English NER. However the SVM classifier outperformed the neural network based model on one dataset (AQMAR). Our proposed models for the Arabic NER outperformed other's proposed models on two Arabic datasets out of three.

The rest of this paper is organized as follows. Related work is discussed in section 2; the datasets and proposed models are presented in the methodology section 3; experimental results and analysis in section 4 and finally the conclusions are discussed in section 5.

2 Related Work

2.1 General NER

There are three main approaches for the NER task: rule-based, machine-learning and hybrid approaches. Early NER approaches were rule-based using handcrafted rules. In rule based approaches, the rules are designed as regular expressions for pattern matching generally with a list of lookup gazetteers [4]. Rule based approaches require expert linguists to design rules for the NER task and usually target a single language. Therefore, few researchers use rule-based systems to develop NER systems [5]. Although the knowledge-based approach can achieve good results, it requires a very exhaustive lexicon in order to work

well. That resulting in inefficiency as entities that don't exist in the lexicon cannot be recognised [6].

There are common classifiers used for NER task such as Conditional Random Fields (CRF), Support Vector Machines (SVM), Maximum Entropy (ME), Decision Trees and Hidden Markov Models (HMM).An important factor in the machine learning based approach is the features that are used. There are some features that have been often used in NER systems such as the case of the word, upper or lower, whether the entity is a digit or contains a digit, and the part of speech associated with a word. The digit feature is useful in NER as it can be used to recognize dates, percentages, money, etc., [7]. The morphology of a word can be captured by including prefixes or suffixes as features. For example, a word can be recognized as an organization if it ended with "tech", "ex" or "soft" [8]. To extract features a window is typically passed over the text. An example of using window feature was proposed by [9] where the part-of-speech of two words before the current word and two words after was used to recognize the named entities. Word length (number of characters) has also found to be an efficient feature for NER task [10].

The third approach to NER, the hybrid approach, which combines both rule-based and machine learning to optimize the system performance [11], In this approach, the output of the rule based system as tagged text is used as input to the machine learning system).

Most of the more recent proposed NER systems are based on recurrent neural networks (RNN) architecture over characters or word embeddings [12]. Those features (word embeddings) are representations of words in n-dimensional space using unsupervised learning over large collections of unlabeled data. The first neural network based approach for NER was proposed by [13]. The system used feature vectors created from orthographic features (e.g., capitalization of the first character), lexicons and dictionaries. Later they replaced these manually created feature vectors with word embeddings. Since then, and starting with [14], implementing neural networks for NER systems have become popular. These kind of models are attractive because they do not require feature engineering efforts, and are thus more domain independent. Current research has shown using pre-trained word embeddings is important for neural network based NER because they are more effective and less time and resource consuming [15]. Also, pre-trained character embeddings is essential for character-based languages such as Chinese (one Chinese character may represent a word meaning) [16].

2.2 Arabic NER

A number of research studies have focused on Arabic Named Entity Recognition ANER. An early attempt for Arabic NER was proposed by [7] where they used a rule-based approach. Their approach consists of a whitelist representing a dictionary of names, and grammar in the form of regular expressions to recognize the named entities. A machine learning-based approach was proposed by [18] where they developed an Arabic NER system named ANERsys 1.0. Linguistic resources have been built by the authors for their experiments including

ANERCorp, the first freely available manually annotated Arabic NER dataset and ANERgazet, an Arabic gazetteer. Contextual and gazetteer features were used in the first version and then part-of-speech features were added in the second version which improved the system performance. A hybrid approach which combines rule-based and machine learning for Arabic NER was proposed by [7]. They used the GATE toolkit [1] for the rule-based approach. The ML-based component used a Decision Tree algorithm. The system used NE tags produced by the rule-based approach besides other language independent features and Arabic specific features.

The missing capitalization feature in the Arabic language is compensated for in some Arabic NER work by using an Arabic morphological analyzer named Buckwalter [33]. Among those features provided by Buckwalter is a feature named English-gloss which provides the English translation for each word in the input Arabic text. Later a tool named MADA was built on Buckwalter and upgraded to be named MADAMIRA [38]. It provides up to 19 orthogonal features. We used some of those features in our designated models which were proven to be efficient in Arabic NER models [38]. More details of the implemented features produced by MADAMIRA are in the features section.

Similar to English, recent work in Arabic NER focuses on developing neural network based approaches. A neural network based approach for Arabic NER employing Bi-LSTM and CRF to predict the named entities has been used [17]. However, their model is missing some techniques such as character representations and hyper parameters tuning. Another approach proposed by [40] used an LSTM neural network model combined with a CNN for character-level features representation. Their model is well designed but is also missing the hyperparameter tuning technique to boost the performance. Also, a new efficient multi-attention technique has been used [41] which uses a combination of word embeddings and character embeddings via an embedding-level attention mechanism. The output is fed into an encoder unit with Bi-LSTM, followed by another self-attention layer to boost the performance.They evaluated their model on ACE and ANERCorp and Twitter datasets. Their model achieved relatively better performance on the ACE dataset which has a different tagging style (not CoNLL-2003 tagging style) and relatively lower performance on Twitter dataset and that is probably due to the noisy text. Their model evaluation is very similar to our neural network based model with a slight inprovement in our results where we are using different hyperparameter values.

Model learning as well as evaluation requires high quality annotated datasets. Initial benchmark datasets were generally created by labeling news articles with a small number of entity types, e.g. CoNLL-2003 [39] and ANERCorp dataset [23]. Later, more datasets were created on numerous kinds of text sources including conversation, Wikipedia articles, and social media such as WNUT-2017 [19]. Arabic datasets are relatively few compared with English datasets and other languages. This represents one of the Arabic NER challenges. Some of widely

[1] https://gate.ac.uk/sale/tao/split.html

used Arabic datasets are ANERCorp created by Benajiba [23] and ACE[2] (commercial dataset).

3 Methodology

In the proposed models, we implemented both traditional machine learning and deep learning for running our experiments and evaluated their performance on different datasets (English and Arabic).

3.1 Datasets

There are five datasets used in the experimental comparison, two English datasets and three Arabic datasets. To cover different datasets aspects (attributes), we adopted diversity in the datasets represented in the different text source of each dataset such as newspapers, Wikipedia and social media. Each dataset is split into training, development and testing sets as indicated in the specified Table 1 for the English datasets and Table 2 for the Arabic datasets. The development set was used for hyperparameter tuning to avoid overfitting.

English Datasets CoNLL-2003: This is a benchmark dataset which was introduced in the Conference on Natural Language Learning (a shared task for named entity recognition) [39] and it has been extensively used in the NER task. The CoNLL-2003 datasets cover several languages and we will focus on the English dataset. The English data was taken from the Reuters Corpus which consists of Reuter's news stories between August 1996 and August 1999. There are four types of named entities in the dataset which are persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC).

WNUT-2017: This high variance dataset was introduced in the Shared Task on Novel and Emerging Entity Recognition 2017 [29]. The named entity tags in this dataset have a wider range including Person, Location (including GPE (Geo Political Entity)), Facility (center, station, etc.), Group (including music band, sports team, and non-corporate organizations), Creative work (song, movie, book, and so on), Corporation and Product (tangible goods, or well-defined services). The source of this dataset is comments taken from social media websites including YouTube comments, Stack Overflow responses, Twitter text for major events in 2016-2017, unfiltered Twitter text 2010, and Reddit comments.

Arabic Datasets ANERCorp: This is a widely used Arabic corpus that was developed by [23] and has the same format as the ConLL-2003 dataset. ANERcorp consists of 316 articles chosen from different newspapers for the sake of generalization. The named entities in this corpus are persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC).

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

AQMAR: This dataset contains 28 hand-annotated Arabic articles collected from Wikipedia with 74,000 tokens [31]. The format of this dataset is similar to CoNLL-2003.

WikiFANEGold: This dataset which is part of dataset named "gold-standard fine-grained NE corpora" was manually created by [32]. The dataset contains Wikipedia articles which were selected by choosing the articles that discuss named entities and considering a fair level of distribution among the classes. In addition, the textual data extracted from the Wikipedia articles was cleaned by removing elements such as headings, lists, and captions on images and tables etc. This dataset consists of 8 coarse-grained classes and 50 fine-grained classes. The coarse grained named entities in this corpus are PER: Person, ORG: Organisation, LOC: Location, GPE: Geo-Political, FAC: Facility, VEH: Vehicle, WEA: Weapon, PRO: Product. We are using coarse grained named entities in our experiments with a total size of 246,303 tokens.

The gazetteers we used in our experiments are ANERgazet [23] for Arabic NER and the English gazetteers used by [30] for English NER which contain lists of persons, locations and organizations names.

Table 1. English Datasets

Dataset	Splits	Tokens	LOC	PER	ORG	MISC	Product	Corp	Creative-Work	Group
CoNLL-2003	Training set	203621	3.5%	3.2%	3.1%	1.7%				
	Development set	51362	3.6%	3.6%	2.6%	1.8%				
	Test set	46435	3.6%	3.5%	3.6%	1.5%				
WNUT-2017	Train set	55725	0.9%	1.0%			0.2%	0.4%	0.2%	0.4%
	Development set	15734	0.5%	3.0%			0.7%	0.2%	0.7%	0.2%
	Test set	8144	1.8%	5.3%			1.6%	0.8%	1.7%	2.0%

Table 2. Arabic Datasets

Dataset	Splits	Tokens	LOC	PER	ORG	MISC	GPE	FAC	VEH	WEA	PRO
WikiFANEGold	Train set	197043	4.0%	34.4%	14.4%		38.2%	2.8%	0.2%	0.3%	5.7%
	Development set	24625	3.9%	34.2%	13.9		37.8%	2.7%	0.3%	0.5%	6.2%
	Test set	24635	5.2%	33.2%	13.8%		37.5%	2.7%	0.6%	0.5%	5.4%
ANERCorp	Train set	12022	29.4%	24.0%	13.5%	7.4%					
	Development set	3150	28.8%	26.3%	16.6%	8.3%					
	Test set	3005	29.5%	24.0%	13.5%	7.4%					
AQMAR	Train set	36050	2.7%	2.1%	0.6%	3.0%					
	Development set	9092	1.8%	2.3%	0.5%	3.9%					
	Test set	9192	1.8%	2.3%	0.6%	4.0%					

3.2 Traditional Machine Learning Based Models

In the traditional machine learning models we implemented supervised machine learning approaches for the NER task as supervised learning approaches out-

perform the unsupervised learning approaches [20]- [23]. A variety of classifiers have been used for NER, however, in our experiments we used Conditional Random Fields (CRF), Support Vector Machine (SVM) and Random Forest (RF) algorithms which have been proven to perform well [10, 21, 22].

3.3 Features

Features to be used in the supervised machine learning approaches were selected based on their performance in other NER research, we used both language-independent and language-specific features. In our proposed models, and in line with previous research, we used the following features which have been proven to be effective:

First, the language-independent features:

- The 3-character-suffix of the word [20]: word suffix information is helpful to identify NEs. This is based on the observation that the NEs share some common suffixes.
- The 3-character-prefix of the word [20].
- Character length of a word [21]: this is a logical valued feature used to check whether the character length of the current word is less than three characters or not. This is based on the observation that the very short words are rarely NEs. If the length of the corresponding word is less than or equal to 3 then the feature values are defined and denoted by False.
- Whether the word contains any digit (0-9) [22]: This feature is helpful in recognizing miscellaneous NEs, such as time expressions, measurement expressions and numerical numbers etc.
- Whether the word contains any punctuation [18].
- Previous NE tag: the previous predicted named entity tag of the current token [22].

Second, the language-specific features:

- Whether the word starts with a capital letter (English only).
- List lookup features (gazetteers) [20]: a set of binary features which capture whether the word is present as a specific entity type in the gazeteer (English and Arabic).
- Part of Speech tags [20]: this feature represents the part of speech tag of the current word and its surrounding words (two previous and two after) (English and Arabic).

  The following are the morphological features generated by MADAMIRA tool [38] for the Arabic NER only:
- Aspect: describes the aspect of an Arabic verb. It has four possible values: Command, Imperfective, Perfective and Not applicable.
- Gender: the nominal Gender. This feature has three values: Feminine, Masculine, Not applicable.
- Person: indicates the person information. The possible values are: 1st, 2nd, 3rd, Not applicable.

– Voice: the verb voice. The values for this feature are: Active, Passive, Not applicable, Undefined, etc.

The baseline model (for both English and Arabic NER) we used included the following features: the 3-character-suffix of the word; the 3-character-prefix of the word; character length of the word; whether the word contains any digit; whether the word contains any punctuation, the Part of Speech tags and capitalization (for English NER only).

3.4 Deep Learning Based Models

In this section, we describe the architecture of the neural network used which is adopted from [24]. The model uses an end-to-end approach that does not require language-specific feature engineering or data pre-processing beyond implementing pre-trained word embeddings. The recognition accuracy can be improved in sequence labeling tasks such as named entity recognition by using the sequence around the word under prediction. Thus, using a Bidirectional Long Short-Term Memory (Bi-LSTM) model can give good performance [24]. The Bi-LSTM model can learn from futuristic and past input features at a particular period of time (e.g. a window approach). The Bi-LSTM model can learn from the past input features using the forward pass technique and the futuristic input features using the backward pass. According to the state-of-the art literature, the Bi-LSTM model can be combined with a Conditional Random Field (CRF) layer to enhance the model performance [24]. The approach of this kind of model is to inherit the ability of learning futuristic and past input features from the Bi-LSTM model and then implement a sentence-level tag to predict the probable tags with the aid of the CRF layer.

In our experiments, we used both word embeddings and character representation as features for our neural network based model (see Fig. 1). To encode character-level information of a word, we used convolutional neural networks (CNNs) for character-level representation. CNNs have been shown to be able to extract morphological information from characters of words [25] such as the word prefix or suffix and encode this information into neural representations. We also choose to use a CNN because we are dealing with the Arabic language. Since Arabic is rich morphological language, using such a technique for Arabic NER will identify different character-level features (word prefixes and suffixes) through the CNN (see Fig. 2). A combination of character- and word-level representations then was fed into a Bi-LSTM. We used a sequential CRF on top of the Bi-LSTM to cooperatively decode labels for the entire sentence.

Employing word embeddings benefits NLP especially when we are dealing with languages that have many rare words and large vocabularies [26], such as Arabic. The nature of the Arabic language, specifically word inflections, generates several lexical variations which lead to sparseness in the Arabic corpus. In our work, for the English NER we used Glove embeddings with 100 dimensions which is publicly available by Stanford, trained on Wikipedia and web text, and contains 6 billion words [27]. For Arabic NER, we used AraVec which is a

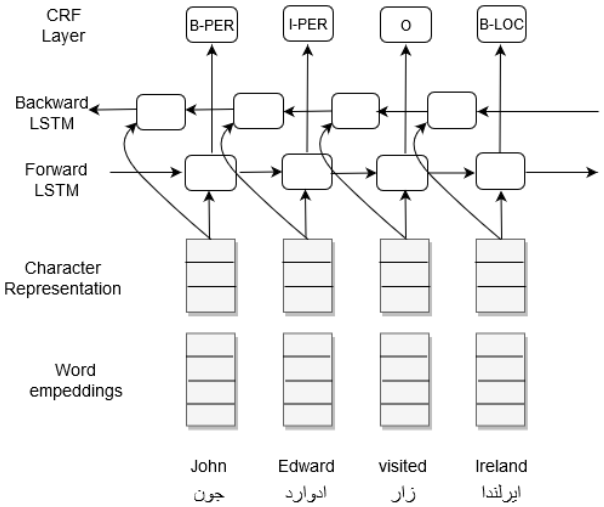

Fig. 1. The general architecture of the neural network based model.

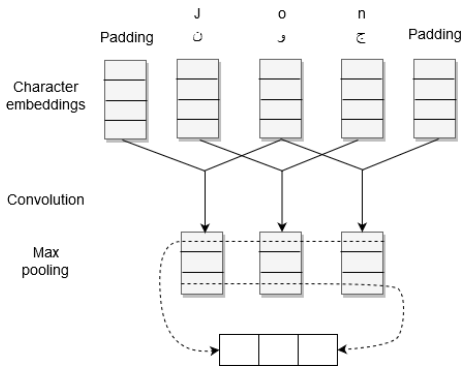

Fig. 2. Character representation using CNN which is then concatenated with the word vector before it is fed into Bi-LSTM.

pre-trained distributed word embedding [28]. AraVec is an open source project which provides free to use Arabic word embeddings trained on more than 3 billion words from web pages and Wikipedia.

3.5 Hyperparameter tuning

In our experiments, we used random search for hyperparameters tuning because it was proven to be more efficient than other tuning approach such as grid search [34]. The hyper parameters identified for the traditional machine learning and the neural network based models are stated in Table 3.

Table 3. Hyper parameters tuning for the traditional ML and the neural network (NN) based models

Classifier	Hyper parameter	Space	Best value
SVM	Kernel	Linear, Poly, rbf, sigmoid	rbf
	C	1e-02 - 1e+03	1e+02
	gamma	1e-02 - 1e+03	1e+01
RF	Number of trees	200 - 2000	400
	Max Features	auto, sqrt	auto
CRF	Optimizer	lbfgs, l2sgd, ap, pa, arow	lbfgs
	Max iterations	100 - 25000	25000
	C1	0.1 – 1.0	0.3
	C2	0.1 – 1.0	0.1
NN	Learning rate	0.005 – 0.008	0.0105
	CNN dropout	0.25 – 0.85	0.25
	Convolution size	3 – 7	3
	LSTM dropout	0.25 – 0.50	0.50
	LSTM state size	100 – 500	200
	Optimizers	SGD, Nadam	Nadam
	Epochs	Determined by performance	120

4 Experimental Results and Discussion

To evaluate our NER models, we used the above mentioned datasets. The CoNLL-2003 dataset splits as training, development and test sets were those offered by the benchmark datasets. The rest of datasets were split into 80% for training, 10% for the development set and 10% for testing. The traditional F-score measure was used to measure the performance. Results for the traditional machine learning approaches are displayed in Table 3. The baseline model for each classifier included the implementation of the classifier with the language-independent features listed in the features section. Since we are focusing on Arabic NER in this paper, we separated the results into two rows for each Arabic dataset. We compared the models performance using two sets of features, the baseline language-independent features labeled as *Sub* and larger feature set, labelled *Full* which included the language-specific features listed in the features section. Baseline results for the three classifiers are displayed in the columns labelled *SVM*, *RF* and *CRF* respectively.

Columns labelled with *-P* show the results of adding in the previous tag because it has been shown that this feature has a big effect on model performance. We excluded this feature in the CRF model as it is, in effect, already included as CRF is a sequential classifier. We also experimented to see the benefit of adding in the use of gazetteers, labelled *-G* in the table. Results show that adding gazetteers also boosted the model performance. The columns labelled as *-PG* include both the previous tag and gazetteers G.

From Table 4 we can reveal some information related to the impact of using previous NE tag and gazetteers features on the performance of each model. On the CoNLL-2003 dataset, the performance of the models improved dramatically

Table 4. The performance of traditional machine learning based models, where *P* means the previous predicted named entity and *G* means Gazetteer. The best performance on each dataset is highlighted in bold.

Dataset	Language	Feature	SVM	SVM-P	SVM-G	SVM-PG	RF	RF-P	RF-G	RF-PG	CRF	CRF-G
CoNLL-2003	English	—	81.23	88.34	85.52	90.13	77.41	85.32	84.32	87.21	91.01	**91.82**
WNUT-2017	English	—	17.42	11.31	19.62	12.32	11.24	08.21	0.12	09.54	33.13	**34.63**
WikiFANEGold	Arabic	Sub	72.35	73.74	74.86	77.57	70.02	70.89	72.23	72.95	77.74	75.63
		Full	73.42	75.26	74.94	77.86	71.85	71.36	72.52	73.62	78.85	**79.24**
ANERCorp	Arabic	Sub	80.21	87.32	82.22	89.45	73.24	82.86	74.48	83.26	75.61	82.40
		Full	83.14	88.95	86.11	**89.81**	75.20	84.34	76.26	85.18	83.09	87.51
AQMAR	Arabic	Sub	73.24	74.46	73.52	74.48	72.16	73.27	73.69	73.75	74.53	74.82
		Full	74.44	75.93	75.21	**76.98**	72.51	73.34	72.28	73.96	74.93	75.89

by using the previous NE tag as well as gazetteers particularly in the SVM model. The best performance on CoNLL-2003 dataset was achieved by using CRF model.

The performance on the WNUT-2017 dataset as we can notice from Table 4 is relatively low due mainly to the noise in the text as it was collected from social networks. Using the previous NE tag or gazetteers on this dataset didn't improve the performance. Instead the performance was decreased probably due to the fact that the previously predicted NE tag is more likely to be wrong which impacts the overall model performance negatively. The same applies to gazetteers - it is difficult to provide support in a gazetteer for such noisy text. However, the CRF model again proved to be relatively successful on this dataset.

In the Arabic datasets, using both language-independent and language-specific features and including the previous predicted NE tag and gazetteers as Full features enhanced the general performance. The best performance was achieved using the SVM model on ANERCorp and AQMAR datasets. The models performance on AQMAR dataset was almost similar to ANERCorp but the performance on WikiFANEGold dataset was lower which is possibly due to the higher number of classes in this dataset. The best performance on this dataset was achieved by using CRF model.

The performance of the Full feature was better than that of the Sub features on all Arabic datasets. However, the effect of using the additional previous NE tag *-P* and gazetteer *-G* is bigger as we notice from Table 4

Table 5 shows the performance of the deep learning based models. The first column in the table labelled *Bi-LSTM* gives performance using only the Bidirectional Long Short-Term Memory algorithm. The second column labelled *Bi-LSTM-CNN* gives the performance of the combination of convolution neural network CNN for character representation and Bi-LSTM. The third column labelled *Bi-LSTM-CNN-CRF* gives performance including the addition of Conditional Random Fields CRF algorithm.

In general, the performance of the deep learning based models is higher than the traditional machine learning based models. Again, the performance on the English datasets is higher than the performance on the Arabic datasets and that is probably due to the challenges with Arabic NER already discussed. Unlike the

Table 5. The F1-score Performance of Deep Learning Mode. The best performance on each dataset is highlighted in bold.

Dataset	Language	Bi-LSTM	Bi-LSTM-CNN	Bi-LSTM-CNN-CRF
CoNLL-2003	English	90.24	91.61	**92.57**
WNUT-2017	English	32.74	35.65	**35.93**
WikiFANEGold	Arabic	78.12	78.9	**79.48**
ANERCorp	Arabic	87.12	89.81	**89.92**
AQMAR	Arabic	75.12	75.52	**76.46**

traditional machine learning based models the differences in performance across the deep learning based models are small.

Table 6 shows the highest performing model from the traditional machine learning approaches and the deep learning based approaches across the Arabic datasets. This table also includes the best performance from existing research on these datasets labelled *SOTA*.

The current best performance on the WikiFANEGold dataset [35] used Buckwalter transliteration, English gloss, POS and NE tag in their model besides window-based and dependency-based representation. They created an approach to capture a global information in the corpus instead of focusing inside the sentence using a CRF classifier in their model. Both the SVM classifier and the deep learning approach used in this paper outperform this approach.

For the ANERCorp dataset the current best performance [36] created a neural network based model which they named Artificial Neural Network (ANN). Their approach included three stages: the first stage was preprocessing the data, in the second they converted Arabic letters to Roman alphabets and in the final stage they applied a neural network to classify their data. They split the dataset into 90% for the training set and 10% for testing set. However, compared to our models, they achieved better performance most probably because of the data pre-processing and converting the Arabic letters to Roman alphabets.

For the AQMAR dataset [37], the authors proposed a model that integrates various custom-made techniques together, including representation learning (a model using word embeddings and Bi-LSTM), feature engineering, sequence labeling, and ensemble learning. They train multiple LSTM-CRF models to construct the mapping from representations to predictions and then concatenate their outputs as ensemble learning. Both the SVM-PG and the deep learning Bi-LSTM-CNN-CRF approach used in this paper outperformed the state of the art on this dataset. Our model SVM-PG gave the best performance on this dataset and this is possibly because in our model, the performance was boosted by using comprehensive language-specific features in addition to the previous NE tag and gazetteers. Also, in our approaches the hyper parameters were tuned using random search technique while it was neglected in both compared models on WikiFANEGold and AQMAR.

Table 6. Comparison between the performance (the highest F1-score) in our traditional ML, Deep learning based models and SOTA

Dataset	SVM-PG	Bi-LSTM-CNN-CRF	SOTA
WikiFANEGold	77.86	**79.48**	73.66 [35]
ANERCorp	89.81	89.92	**92.36** [36]
AQMAR	**76.98**	76.46	75.82 [37]

5 Conclusion

In this paper, we have explored a variety of different approaches to NER on Arabic text with reference to how these approaches perform also on English text. The exploration involves evaluating different classifiers and features on a number of datasets. The selected datasets are diverse in terms of contents source (e.g. news articles, twitter, etc.). We evaluated both language specific and language independent features. We found that adopting the language specific features and using gazetteers and the previous predicted named entity tag can achieve higher performance in traditional machine learning based models. Also, the deep learning based models have higher performance evaluations on the most of datasets. Our proposed models outperformed the related work on two Arabic datasets out of three. However, the performance on the English datasets are higher than the Arabic datasets because of the characteristic of the Arabic language represented in the morphological ambiguity.

References

1. Nadeau, David, and Satoshi Sekine. "A survey of named entity recognition and classification." Lingvisticae Investigationes 30, no. 1 (2007): 3-26.
2. Habash, Nizar Y. "Introduction to Arabic natural language processing." Synthesis Lectures on Human Language Technologies 3, no. 1 (2010): 1-187.
3. Farber, Benjamin, Dayne Freitag, Nizar Habash, and Owen Rambow. "Improving NER in Arabic Using a Morphological Tagger." In LREC. 2008.
4. Rau, Lisa F. "Extracting company names from text." In [1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application, vol. 1, pp. 29-32. IEEE, 1991.
5. Gaizauskas, Robert, Takahiro Wakao, Kevin Humphreys, Hamish Cunningham, and Yorick Wilks. "UNIVERSITY OF SHEFFIELD: DESCRIPTION OF THE LaSIE SYSTEMAS USED FOR MUC-6." In MUC-6:, November 6-8, 1995.
6. Segura Bedmar, Isabel, Paloma Martínez, and María Herrero Zazo. "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)." ACL, 2013.
7. Shaalan, Khaled, and Hafsa Raza. "Arabic named entity recognition from diverse text types." In Int Conf on NLP, pp. 440-451. Springer, Berlin, Heidelberg, 2008.
8. Bick, Eckhard. "A Named Entity Recognizer for Danish." In LREC. 2004.
9. Chieu, Hai Leong, and Hwee Tou Ng. "Named entity recognition: a maximum entropy approach using global information." In Proceedings of the 19th Int Conf on Computational linguistics-Volume 1, pp. 1-7. ACL, 2002.

10. Abdul-Hamid, Ahmed, and Kareem Darwish. "Simplified feature set for Arabic named entity recognition." In Proceedings of the 2010 Named Entities Workshop, pp. 110-115. ACL, 2010.
11. Petasis, Georgios, Frantz Vichot, Francis Wolinski, Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos. "Using machine learning to maintain rule-based named-entity recognition and classification systems." In Proceedings of the 39th Annual Meeting on ACL, pp. 426-433., 2001.
12. Kim, Yoon, Yacine Jernite, David Sontag, and Alexander M. Rush. "Character-aware neural language models." In Thirtieth AAAI Conference on Artificial Intelligence. 2016.
13. Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of machine learning research 12, no. Aug (2011): 2493-2537.
14. Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." In Proceedings of the 25th international conference on Machine learning, pp. 160-167. ACM, 2008.
15. Habibi, Maryam, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. "Deep learning with word embeddings improves biomedical named entity recognition." Bioinformatics 33, no. 14 (2017): i37-i48.
16. Yin, Rongchao, Quan Wang, Peng Li, Rui Li, and Bin Wang. "Multi-granularity chinese word embedding." In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 981-986. 2016.
17. Sa'a, D. A. Alzboun, Saia Khaled Tawalbeh, Mohammad Al-Smadi, and Yaser Jararweh. "Using bidirectional long short-term memory and conditional random fields for labeling arabic named entities: A comparative study." In 2018 Fifth Int Conf on Social Networks Analysis, Management and Security (SNAMS), pp. 135-140. IEEE, 2018.ties.
18. Benajiba, Yassine, and Paolo Rosso. "ANERsys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-tag Information." In IICAI, pp. 1814-1823. 2007.
19. Derczynski, Leon, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 shared task on novel and emerging entity recognition." In Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 140-147. 2017.
20. AbdelRahman, Samir, Mohamed Elarnaoty, Marwa Magdy, and Aly Fahmy. "Integrated machine learning techniques for Arabic named entity recognition." IJCSI 7 (2010): 27-36.
21. Abdul-Hamid, Ahmed, and Kareem Darwish. "Simplified feature set for Arabic named entity recognition." In Proceedings of the 2010 Named Entities Workshop, pp. 110-115. Association for Computational Linguistics, 2010.
22. Ekbal, Asif, and Sivaji Bandyopadhyay. "Named entity recognition using support vector machine: A language independent approach." International Journal of Electrical, Computer, and Systems Engineering 4, no. 2 (2010): 155-170.
23. Y. Benajiba, P. Rosso, and J. M. Benedíruiz, "Anersys: An arabic named entity recognition system based on maximum entropy," in Int Conf on Intelligent Text Processing and Computational Linguistics, 2007, pp. 143–153.
24. Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural architectures for named entity recognition." arXiv preprint arXiv:1603.01360 (2016).
25. Chiu, Jason PC, and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs." Transactions of the ACL 4 (2016): 357-370.

26. Zirikly, Ayah, and Mona Diab. "Named entity recognition for arabic social media." In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 176-185. 2015.
27. Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
28. Soliman, Abu Bakr, Kareem Eissa, and Samhaa R. El-Beltagy. "Aravec: A set of arabic word embedding models for use in arabic nlp." Procedia Computer Science 117 (2017): 256-265.
29. Derczynski, Leon, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 shared task on novel and emerging entity recognition." In Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 140-147. 2017.
30. Nadeau, David, Peter D. Turney, and Stan Matwin. "Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity." In Conference of the Canadian society for computational studies of intelligence, pp. 266-277. Springer, Berlin, Heidelberg, 2006.
31. Mohit, Behrang, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. "Recall-oriented learning of named entities in Arabic Wikipedia." In Proceedings of the 13th Conf of the European Chapter of the ACL, pp. 162-173., 2012.
32. Alotaibi, Fahd, and Mark Lee. "A hybrid approach to features representation for fine-grained Arabic named entity recognition." In Proceedings of COLING 2014, the 25th Int Conf on Computational Linguistics: Technical Papers, pp. 984-995. 2014.
33. Buckwalter, Tim. "Issues in Arabic orthography and morphology analysis." In proceedings of the workshop on computational approaches to Arabic script-based languages, pp. 31-34. ACL, 2004.
34. Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 13, no. Feb (2012): 281-305.
35. Alotaibi, Fahd, and Mark Lee. "A hybrid approach to features representation for fine-grained arabic named entity recognition." In Proceedings of COLING 2014, the 25th Int Conf on Computational Linguistics: Technical Papers, pp. 984-995. 2014.
36. Mohammed, Naji F., and Nazlia Omar. "Arabic named entity recognition using artificial neural network." Journal of Computer Science 8, no. 8 (2012): 1285.
37. Liu, Liyuan, Jingbo Shang, and Jiawei Han. "Arabic Named Entity Recognition: What Works and What's Next." In Proceedings of the Fourth Arabic NLP Workshop, pp. 60-67. 2019.
38. Pasha, Arfath, Mohamed Al-Badrashiny, Mona T. Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic." In LREC, vol. 14, pp. 1094-1101. 2014.
39. Sang, Erik F., and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." arXiv preprint cs/0306050 (2003).
40. Khalifa, Muhammad, and Khaled Shaalan. "Character convolutions for Arabic Named Entity Recognition with Long Short-Term Memory Networks." Computer Speech and Language 58 (2019): 335-346.
41. Khalifa, Muhammad, and Khaled Shaalan. "Character convolutions for Arabic Named Entity Recognition with Long Short-Term Memory Networks." Computer Speech and Language 58 (2019): 335-346.