Arabic Named Entity Recognition using Clustered Word Embedding

Caroline Sabty, Mohamed Elmahdy and Slim Abdennadher

Computer Science and Engineering Department,
The German University in Cairo
Cairo, Egypt
{caroline.samy, mohamed.elmahdy, slim.abdennadher} @guc.edu.eg

Abstract. Named Entity Recognition in Arabic is a challenging topic because of morphological and lexical richness of Arabic. In this paper, we propose an Arabic NER system that is based on word embedding. Word embedding hold semantic information about the context of the words. We hypothesized that the integration of word embedding features to the conventional lexical and contextual features could improve Arabic NER performance. The Conditional Random Field (CRF) sequence classifier was used. Since most CRF implementations only support categorical features, continuous word embedding vectors are clustered. In this paper, we are mainly investigating the effect of the number of clusters on NER performance. Moreover, the combination of fine and coarse grained clusters has resulted in further recognition improvement.

Key words: Named Entity Recognition, Arabic, Conditional Random Field, Clustering, Word Embedding, Natural Language Processing

1 Introduction

Arabic Natural language processing (NLP) has gained increasing importance and a lot of research has been conducted for various ranges of applications such as Name Entity Recognition (NER). It is the process of extracting and classifying the named entities from an unstructured text into a set of predefined classes [1].

One of the main statistical Machine learning techniques is Conditional Random Field (CRF). CRF is a sequence classifier to segment and label sequence data based on probabilistic models. It could be considered as an enhancement or generalization of Maximum Entropy (ME) and Hidden Markov Models (HMM) [2]. Recently, CRF has shown to be very successful in different NLP tasks and specially in NER [3]. One of the main advantages of CRF is the ability to consider contextual information before assigning a label to a word. N-gram algorithm and other available NLP techniques consider words as atomic units that do not have anything in common, which results in simplicity and capability to train big amount of data. However, these techniques recognize mostly words that are available in the training data [4]. Moreover, [5] proved that using the concept of distributed representations of words like neural network based language models

is better than N-gram models. The word representation (embedding) is a fixed size vector that capture the morphological and semantic information of the word.

Arabic is a morphologically rich language. As a lot of words have different prefixes and suffixes, which makes it hard to identify the similarity between them. However, with the usage of word embedding it is easy to find these similarities as similar words are mapped to nearby vectors. For instance, the words 'الدولة' (the country), 'الدولتان' (the two countries), 'الدول' (the countries) differ in their morpholical forms, however, their embedding should be placed near to each other in the space. Arabic is also a lexically rich language, different words correspond to the same sense. For example, the two Arabic words 'دولة' and 'بلد' correspond to the same English meaning: "country".

One of the aims of this work is to show how unsupervised word embedding with CRF can be integrated to perform Arabic Named Entity Recognition task. This combination could not be done directly as CRF systems only allows categorical features and not continuous features such as word embedding vectors. The proposed solution is to cluster the generated vectors and plug the generated cluster IDs in the feature vector of the CRF system along with other lexical and contextual features. In order to cluster the generated vectors, we used K-means clustering. However, it is hard to know the optimal number of clusters. So, we compared different numbers of clusters to know which one gets the best performance. Our evaluation demonstrates the effectiveness of word embeddings clusters along with CRF for Arabic NER task.

In addition, some Arabic words can have different labels or NE types based on the context of the word. Thus, investigating the different contextual features for a given word might lead to a differently classified type. However, the question is how long should we consider in the left and right context of the word. A lot of studies have been conducted in this area and no final decision was taken, so we decided to evaluate different combinations to get the optimal context settings.

The paper is organized as follows. Section 2 presents some related work. In Section 3, the system design is illustrated by presenting the ANER pipeline, baseline features and word embeddings features. The evaluations and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Due to the morphological complexity of the Arabic language, few attempts have been done tackling the problem of NER in Arabic compared to other languages such as English. Two main approaches are being used in Arabic NER systems: the rule-based (RL) and machine learning (ML) based approaches [6]. The RL NER systems depends on grammar rules, usually represented as regular expressions. ML NER systems use learning algorithms that need large tagged training and testing data and use sets of features from these annotated datasets. Regarding the RL approach, it has been used by [7] for Arabic NER task. The system relied on a whitelist containing names and a set of grammar rules. It was evaluated using their own data set, the results of the evaluation accomplished high

F1-measure of 85.9% for Location, 87.7% Person and 83.15 % for Organization. However in general, the RL approaches need a strong linguistic knowledge and they are time consuming.

Concerning the ML approaches, [8] used the Maximum Entropy (ME) and n-grams based algorithms for Arabic NER. They built a system (ANERsys) and they created corpus (ANERcorp) and gazetteer (ANERgazet) to training and testing. The system achieved 55.23% F1-measure. They enhanced the system in [9] by comparing two different techniques, Support Vector Machines (SVM) and CRF. In addition, they explored different combination of features such as contextual, lexical and morphological on different data sets. They could not state which one SVM or CRF is better as it differs for different entity types. The best results they achieved were 83.5% F1-measure for the ACE 2003 BN data. At the end in [3] they replaced the probabilistic model from ME to CRF and they got as a result 79.21% F1-measure. A set of features is being investigated in [10] to be used for CRF sequence labeling. They showed that character n-grams of leading and trailing characters of a word can be represented as a lexical features. This could help in the NER task without the use of linguistic analysis. They achieved 81% F1-measure on Benajiba dataset [8] and 76% F1-measure on ACE 2005 dataset. However, they have only considered three NE types (persons, locations and organizations).

Lately, a combination of both RL-based and ML-based approaches is used as a hybrid approach. For instance, in [11] they used a hybrid approach for Arabic NER. The RL part in their system is similar to the one presented in [7]. Regarding the ML part started by features and classifiers selections. Their approach showed promising results, however, it still has the problems of the RL approaches. The system performance was 90.1% F1-measure for Location, 94.4% for Person and 90.1% for Organization. Other available approaches has been used for Arabic NER such as using a leveraging parallel corpora (Arabic-Spanish) and previously developed tools for other language (Spanish) [12].

Few attempts have been done towards using word embedding for NER tasks in general. [13] Up to our knowledge, for Arabic language, only two systems used word embedding combined with CRF for Arabic NER. The first one is presented in [14], it showed promising results, however, it is designed to recognize entities extracted from social media text written in dialectal Arabic. It achieved 72.68% F1-measure on a dialectal dataset. In addition, few details are mentioned concerning the dimensions of the generated vectors and numbers of clusters. The second one presented in [15] mainly focused on comparing between two different word embedding algorithms (Word2Vec and Global Vectors). They used AQMAR Corpus which consists of 74k token. The best performance their system achieved was 67.22% F1-measure. Nevertheless, they did not study the effect of number of clusters.

3 Proposed System Design

The proposed ANER system implemented recognizes three different types of entities: Location (LOC), Person (PERS) and Organization (ORG). In addition, it tags entities that do not belong to these types with miscellaneous (MISC) and the ones that are not considered NE are tagged with other (O). The data used in the training and testing is the ANERCorp created by [8]. Labels follows the IOB format (classes) that is used in MUC-6 tasks[1]. Each class has two types: B-class and I-class. The B-class denotes the beginning of an entity and the I-class denotes the inside of a class. This data set is chosen because it is the largest free annotated corpus for ANER. It consists of 150,286 tokens and 32,114 labeled NE.

3.1 ANER Pipeline

As shown in Figure 1, the system starts by normalizing the data. As the Arabic letters have different shapes, normalization process was needed to unify some letters that are written differently. For instance, 'آ' and 'أ' are replaced with 'ا' and some punctuations such as '.' and ',' were removed. The word2Vec (W2V) model was trained using an independent Arabic newswire data-set. The normalized training data and the model were used to generate vectors for the training data. After, the vectors are clustered using a clustering model. The output from the previous step is used to generate IDs based on the cluster number for the training data. These IDs were added as features in the CRF system. Several features were added to the training data as will be explained later. In the final step, the new generated training data with all the features are fed as inputs to the CRF algorithm. The toolkit used to apply the CRF algorithm is CRF++ [16]. It is an open source CRF tool, used mainly for sequence classification. The main advantages of this tool is the ability to handle large feature sets and has the option of using multi-threading option which makes it much faster than other existing CRF toolkits.

3.2 Initial Baseline Features

Features are considered the characteristic attributes of words that should be used with ML algorithms [1]. The main selected features are: stemming, POS tagging, and some contextual and lexical features.

3.2.1 Stemming Features In order to get the single representation of the words which is called Stem, a new approach should be implemented or used [17]. Adding the stems of the words as a feature to the CRF algorithm can help matching similar words with different morphological representations. Thus, we used the "ISRI stemmer" algorithm that is describes in [17] for word stemming on the whole dataset.

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

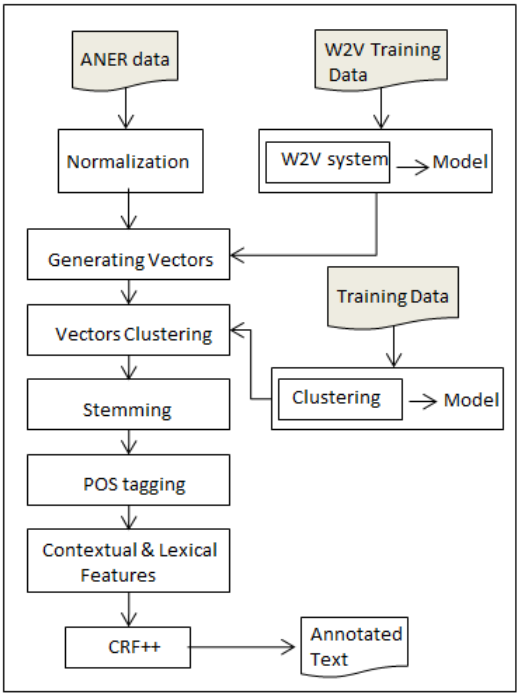

Fig. 1. A block diagram for the proposed system

3.2.2 Part-Of-Speech tag Features Several POS tagging tools were investigated and the one that resulted in the best tagging performance is RDRPOSTagger. It is a language independent tool, that identifies part-of-speech tags (e.g., Nouns, Verbs). RDRPOSTagger applies an error-driven approach to construct a Single Classification Ripple Down Rules tree of transformation rules [18].

3.2.3 Contextual and Lexical Features Contextual features are automatically generated based on the context of the NE. The context can be any number of previous or next words. We investigated several sets of features for this class of Contextual features to get the optimal settings. Concerning the lexical features, they are the character n-grams of the tokens, in other words, it could be considered as the fixed length prefix and suffix of a word. In our system the first and last letters of a word are added as two lexical features.

3.3 Word Embeddings based Feature

In this work, Word2Vec is the algorithm used to produce the word embedding vectors. It is based on deep learning. And it uses skip-gram and Continuous

Bag-of-Words (CBOW) models that is presented in [4]. CBOW predicts the probability of a word in a context. Whereas, skip-gram predicts the word based on a given context. W2V provides multiple degrees of similarity between different words by mapping to nearby vectors, which is very useful for Arabic language. A dataset is used to train the W2V model, that consists of 84M words.

The generated embeddings vectors are indirectly integrated in our system by adding them as a feature to the CRF algorithm to recognize Arabic NE. In order to convert continuous vectors into categorical features a clustering technique is used to cluster the vectors and adding their clustering IDs as a features. K-means is the clustering algorithm used. Two of the main factors that affect the performance of the system are the vector size and the number of clusters. The vector size chosen is 100. We have investigated several numbers of clusters as shown in Section 4.

4 Evaluation and Results

We divided the ANERCorp corpus into training and testing data of 110,286 and 40k tokens respectively. The evaluation process was divided into four experiments. First the best combination of contextual features was checked. Second, the performance of the system was evaluated by adding different features to build the baseline. The third experiment integrated the word embedding with the selected set of features and compared the different number of clusters for the word embedding. Finally, combining the coarse and fine grained clusters IDs was investigated.

4.1 Contextual Features

The first experiment started by evaluating the baseline with only the current word as a feature. Then, the next and the previous words were added to the current word separately. Furthermore, we have combined left and right contexts together. In the combination the left or right context consists of either one or two words. The performance measures used in our evaluation are precision (P), recall (R) and F-measure (F1).

In Table 1, the results of the different contextual features are listed. According to the Table 1, the best setting achieved by using the current and the previous word, they achieved 59.7% F1-measure. Whereas, the lowest results came from using only the current and the 2 next words 54% F1-measure.

4.2 Baseline Features Set

In Table 2, the performances of the CRF system using the different types of features are illustrated. The figure shows an increase in the value of the F1-measure by appending the features together. For instance, the addition of the lexical feature to the word stem and the current word increased the F1-measure from 64.1% to 66%. Moreover, by adding all the features to build the baseline F1-measure increased to 68.4%.

Feature	P	R	F1
Current	97	41.7	58.3
Current-1Next	96.3	37.9	54.3
Current-2Next	95.5	35.3	51.5
Current-1Previous	98	43	59.7
Current-2Previous	97.1	40	56.6
Current-Previous-Next	96	40	56.4
Current-2Previous-2Next	96.9	37.5	54

Table 1. The performance measures results in (%) from using different combinations of contextual features

Feature	P	R	F1
Current-Stemming	96	48.2	64.1
Current-Stemming-Lexical	94.2	50.9	66
Current-Stemming-Lexical-Contextual	95.1	51.2	66.5
Current-Stemming-Lexical-Contextual-POStag	93.2	54.1	68.4

Table 2. The performance measures results in (%) using different set of features

4.3 Cluster Granularity

In order to add the cluster ID of the word embedding as a feature, an experiment was done to get the relevant cluster granularity. Several number of clusters were experimented to get the cluster size with the best performance, the result of this experiment is shown in Table 3

The evaluation started by adding the IDs of a small size cluster 50 to the set of features of the baseline. We kept increasing the size of the cluster and the performance increased as well till reaching the cluster with size 500 and after that the performance started to decrease. The figure indicates that the cluster with size 50 is the one with the worst results of 72.7% and the cluster with size 500 is the one with the best result of 76.1% F1-measure. Thus, adding the IDs of the cluster 500 to the feature set enhanced the performance of the baseline from 68.4% to 76.1%.

Furthermore, we investigated combining the coarse and fine grained clusters by adding the IDs generated by the number of clusters 50 and 500 to the feature set. The output demonstrated that the performance has slightly increased to 76.4% F1-measure. We interpret the improvement after the combination as coarse grained clustering could help more in modeling rare words. On the other hand, fine grained clustering can results in better performance for common words.

No. of Clusters	P	R	F1
50	90	61.1	72.7
100	90	62.6	73.8
200	90.7	64.2	75.1
500	92	64.9	76.1
1,000	90.6	64.1	75
4,000	90.6	63.7	74.8
50 & 500 combined	91.4	65.7	76.4

Table 3. The performance measures results in (%) using different number of clusters

5 Conclusion

An effective integration between CRF and word embedding for Arabic Named Entity Recognition task was presented in this paper. The benefits of using word embedding in Arabic NER task were illustrated. Word embedding can be integrated in CRF classifier by clustering the vectors and adding the cluster ID as a feature. Different number of clusters were investigated and 500 clusters achieved the highest performance, while 50 clusters got the lowest performance results. The best word embedding settings of combining fine and coarse cluster IDs results in 76.4% F-measure with a relative improvement from the baseline of 11.7%. Thus, we can conclude that the system achieved the best performance by the following set of features: Current word, Stemming, Lexical, Contextual, POS tagging, fine and coarse word embedding cluster IDs.

References

1. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30** (2007) 3–26
2. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001) 282–289
3. Benajiba, Y., Rosso, P.: Arabic named entity recognition using conditional random fields. In: Proc. of Workshop on HLT & NLP within the Arabic World, LREC. Volume 8. (2008) 143–153
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
5. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research **3** (2003) 1137–1155
6. Shaalan, K.: A survey of arabic named entity recognition and classification. Computational Linguistics **40** (2014) 469–510
7. Shaalan, K., Raza, H.: Nera: Named entity recognition for arabic. Journal of the Association for Information Science and Technology **60** (2009) 1652–1663
8. Benajiba, Y., Rosso, P., Benedíruiz, J.: Anersys: An arabic named entity recognition system based on maximum entropy. Computational Linguistics and Intelligent Text Processing (2007) 143–153

9. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition using optimized feature sets. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2008) 284–293
10. Abdul-Hamid, A., Darwish, K.: Simplified feature set for arabic named entity recognition. In: Proceedings of the 2010 Named Entities Workshop, Association for Computational Linguistics (2010) 110–115
11. Oudah, M., Shaalan, K.F.: A pipeline arabic named entity recognition using a hybrid approach. In: Coling. (2012) 2159–2176
12. Samy, D., Moreno, A., Guirao, J.M.: A proposal for an arabic named entity tagger leveraging a parallel corpus. In: International Conference RANLP, Borovets, Bulgaria. (2005) 459–465
13. Seok, M., Song, H.J., Park, C.Y., Kim, J.D., Kim, Y.s.: Named entity recognition using word embedding as a feature. International Journal of Software Engineering and Its Applications **10** (2016) 93–104
14. Zirikly, A., Diab, M.T.: Named entity recognition for arabic social media. In: VS@ HLT-NAACL. (2015) 176–185
15. Laachfoubi, N., et al.: Arabic named entity recognition using word representations. International Journal of Computer Science and Information Security **14** (2016) 956
16. Robert Parker, David Graff, K.C.J.K., Maeda, K.: Arabic gigaword fourth edition ldc2009t30. Linguistic Data Consortium (LDC), Philadelphia (2009)
17. Taghva, K., Elkhoury, R., Coombs, J.: Arabic stemming without a root dictionary. In: Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on. Volume 1., IEEE (2005) 152–157
18. Nguyen, D.Q., Nguyen, D.Q., Pham, D.D., Pham, S.B.: Rdrpostagger: A ripple down rules-based part-of-speech tagger. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. (2014) 17–20