International Journal of Transdisciplinary AI
Vol. 1, No. 1 (2019) 44-63

DOI: 10.35708/TAI1868-126245

Techniques for Named Entity Recognition on Arabic-English Code-Mixed Data

Caroline Sabty, Ahmed Sherif, Mohamed Elmahdy, and Slim Abdennadher

Computer Science and Engineering Department,
The German University in Cairo,
El Tagamoa El Khames, New Cairo, Cairo, Egypt
caroline.samy@guc.edu.eg
met.guc.edu.eg

Received (07/18/2019)
Revised (08/20/2019)
Accepted (09/30/2019)

Abstract. As a result of globalization and better quality of education, a significant percentage of the population in Arab countries have become bilingual/multilingual. This has raised the frequency of code-switching and code-mixing among Arabs in daily communication. Consequently, huge amount of Code-Mixed (CM) content can be found on different social media platforms. Such data could be analyzed and used in different Natural Language Processing (NLP) tasks to tackle the challenges emerging due to this multilingual phenomenon. Named-Entity Recognition (NER) is one of the major tasks for several NLP systems. It is the process of identifying named entities in text. However, there is a lack of annotated CM data and resources for such task. This work aims at collecting and building the first annotated CM Arabic-English corpus for NER. Furthermore, we constructed a baseline NER system using deep neural networks and word embeddings for Arabic-English CM text. Moreover, we investigated the usage of different types of classical and contextual pre-trained word embeddings on our system. The highest NER system achieved an F1-score of 77.69% by combining classical and contextual word embeddings.

Keywords: Named Entity Recognition; Arabic Language; Code-Mixed; Word Embedding; Deep Neural Network.

Introduction

The Arabic language has gained a lot of attention in computational linguistics. Arabic exists in different forms such as Modern Standard Arabic (MSA) and

dialectal Arabic (DA). MSA is mostly used in formal context. However, DA is used more in everyday life communications (written or spoken) between Arabic speakers. There are several types of dialectal Arabic in different countries e.g. Egyptian, Levantine, Gulf, Iraqi, etc. Within each country, native speakers tend lately to use code-switching (CS) and code-mixing (CM) and they exchange between their own dialect and other languages. For example, in Egypt, people tend to code-mix with English; in Algeria, they tend to code-mix with French. CS or CM refers to the behaviour of mixing more than one language in the same context or conversation. It is a common behaviour that happens in written and spoken languages.

- CS is defined as switching between different languages from one sentence to another [11]. For example:
  That's a great idea! ممكن نخرج بكرا.
  (We can go out tomorrow. That's a great idea!)

- CM refers to using multiple languages within the same sentence [11]. For example:
  كأس السوبر السعودي الهلال vs الاتحاد يوم Saturday في London.
  (The Saudi Super Cup Al Hilal vs Al Aitihad on Saturday in London.)

Code Mixing has become a common behaviour in multilingual communities especially on social media platforms. Thus, there is a huge need to process and analyze such data in order to be used in different Natural Language Processing (NLP) tasks.

One of the main important tasks for NLP is Named-Entity Recognition (NER). NER is the task of identifying named entities and classifying them into different types or categories such as persons, locations and organizations. Much work has been conducted in this task for some major languages, such as English. However, there is less work done for Arabic and no work done for Arabic-English CM data. In this paper, we refer to the Egyptian dialect along with English code-mixing by Arabic-English CM data. In general, the state-of-the-art algorithms have less accuracy on informal text such as the text generated from social media platforms, e.g. Twitter [17]. This is due to the limitation of the generated and annotated text such as the short length, the informal languages used, and the CM behaviour in the text.

Recently, Deep Neural Network (DNN) models proved to be effective in solving a wide range of NLP tasks including NER [27]. The main advantage of using DNN techniques is the ability of the models to learn all features from the data without having them set in advance. In addition, DNN has the capability of using word embeddings. This refers to representing each word as a vector which results in having semantically similar words becoming near each other in space. Adding word embedding is very useful in estimating semantic information about the data [30].

However, classical word embeddings do not take into consideration the context of the words. For instance, the word *"apple"* has a totally different meaning

in the following two sentences *"I want to eat an apple"* and *"Apple store is very crowded today"*. Thus, using classical word embeddings will generate the same vector for the word "apple" in both sentences. Therefore, in order to overcome this ambiguity problem contextual word embeddings are being introduced and several types are being proposed in the NLP field. Contextual word embeddings generate for the same word different embeddings based on its context, which is essential to capture the semantics of the ambiguous words based on their context [35]. In this example "apple" will be given different vectors in both sentences. This will help identifying that in the first sentence it refers to a fruit and in the second one to a store or organization.

This paper is an extended and revised version of [38], where we collected a code mixed corpus for Named Entity Recognition on Arabic-English text and proposed a DNN-based NER baseline system and a pooling technique for this kind of CM data. To the best of our knowledge, we present the first collected and annotated corpus for code-mixed Arabic-English data for NER tasks. In addition, no work has been conducted in this direction for the task of NER for Arabic-English CM text. The corpus is concerned mainly with the script of the Egyptian Arabic dialect for the time being. Three techniques have been applied to combine and develop the corpus. The first one was using data harvested from Twitter. The second one was translating some of the existing annotated Arabic NER data. The last technique was using data from the transcribed speech corpus of [23]. After gathering the data, named entities were annotated.

The main architecture of our deep neural network models is Bidirectional Long-Short-Term-Memory (BiLSTM) and Conditional Random Field (CRF) along with pre-trained word embedding layer.

In order to build a CM baseline, conventional training techniques were adopted. This was done by training two different models: one for Arabic NER and another one for English. The Arabic model was trained using the two corpora: ANERCorp [7] and AQMAR [31]. The English model was trained using the CoNLL 2003 corpus [44]. The baseline system relies on detecting the language of the testing words and based on it, the predicted tag is taken from either the Arabic or English predictions. The performance of the baseline system was 52% F1-score.

Previously, in order to improve the baseline we proposed a pooling technique, which was done by combining the same Arabic and English NER corpora of the baseline to form the training data-set. Training is done once in this case. The model loads two classical pre-trained word embedding models; one for English and one for Arabic text. The pooling system reached a performance of 60% F1-score.

In this work, we have extended our collected code-mixed corpus for NER on Arabic-English data. The developed corpus increased from 1,331 to 6,525 sentences. The number of sentences containing entities is equal to 4,948 sentences which is 75.8% of the total number of sentences.

Currently, after collecting more CM data we were able to train and test different models with CM data. At the beginning, the same model used in the pooling technique was trained with the newly collected CM data and tested with

the same testing data. It achieved a higher absolute F1-score of 6.04% from the one of the pooling [38].

However, in order to enhance the performance and to have an NER system with higher results we tried the different state-of-the-art classical and contextual word embeddings with the same architecture of BiLSTM-CRF. We were able to significantly improve the performance of the system by 25.69% absolute F1-score from the baseline to reach 77.69% F1-score by classical and contextual word embeddings.

The paper is organized as follows. Section 2 presents some related work for NER and CM. In Section 3, the process of data collection and annotation is illustrated. Section 4 explains the different types of pre-trained word embeddings. Section 5 presents the BiLSTM-CRF architecture of the model. The experiments and results are discussed in Section 6. Finally, Section 7 concludes the paper and presents future work.

Related Work

In this section, we provide an overview on some of the previous work related to NER and code-mixing.

Named Entity Recognition

A lot of research has been conducted on NER for monolingual text. Different machine learning approaches such as Hidden Markov Models, Conditional Random Fields (CRF), Support Vector Machines (SVM) and decision trees for NER achieved high performance results on different languages. For instance, one of these approaches applied on Arabic data for NER is presented in [8]. They built a system for Arabic NER using Maximum Entropy Markov model and n-grams and created the ANERcorp Corpus. The system achieved 55.23% F1-score. They enhanced their results in [7] by comparing SVM and CRF techniques and got 83.5% F1-score. Furthermore, [12] proposed Arabic NER system that is based on word embedding and CRF. They clustered the word embedding and added the clustering IDs as a feature along with a set of conventional lexical and contextual features. The system achieved 76.4% F1-score on ANERCorp [8] and AQMAR [31] Corpus.

Supervised or semi supervised machine learning approaches require domain specific resources and a lot of feature engineering. That is why deep sequence classifiers like Recurrent Neural Network (RNN) systems have been proposed for NER task and improved performance significantly [46]. The first word embedding model implemented along with neural network system in [14] showed the importance of having word embedding in several NLP tasks such as Named entity Recognition. For NER, the paper used a convolution layer that was connected to a CRF layer, and achieved 89.59% F1-score on the English CoNLL 2003 data-set. [47] introduced different neural network based models for NER on three data-sets in English, German and Arabic. For the English they also used the CoNLL 2003 data-set, for German they used GermEval 2014 NER shared task [9] and for the Arabic they used ANERcorp. The different models they experimented were BiLSTM, window BiLSTM and a word level feed-forward. They

also added different features such as CRF, Part-of-Speech (POS) tagging and word embedding. The best results they achieved for English was 88.9%, German was 76.1% and for Arabic 71.3% F1-score.

In addition, [27] created two models using BiLSTM and added output label dependencies via CRF and another one using a transition-based approach inspired by a shift-reduce parser for NER on four languages. They achieved the best performance using the BiLSTM and CRF model. The results they got are considered one of the highest compared to other state-of-the-art systems. For the English and German languages they used CoNLL 2003 data-set and achieved 90.94% and 78.76% F1-score. Moreover, for the Dutch and Spanish languages they used CoNLL 2002 data-set [44] and achieved 81.84% and 85.75% F1-score. Moreover, [16] used deep learning techniques for Arabic NER. They combined different techniques such as character- and word-level representations, BiLSTM, CRF and Convolutional Neural Network (CNN) to build their NER model. They also tested the effect of different hyper-parameters on the final performance. The best model obtained after tuning the hyper parameters got an F1-score of 76.65%.

As traditional word embeddings do not take into consideration the context of the words, recently new types of contextual word embeddings are being proposed and used in several NLP tasks. Some work has been conducted for NER task using contextual embeddings for English language. In [34] they introduced a semi-supervised approach using bidirectional language models for adding contextual pre-trained embeddings to different NLP tasks. They evaluated their model using CoNLL 2003 English data-set for NER. It achieved an F1-score of 91.93% which is an improvement of 1% from the baseline system that was implemented with the normal pre-trained embeddings.

Moreover, [35] proposed a new type of embeddings which is a deep contextualized word embeddings (ELMo) that could be added to existing models. It considers the syntax and semantics of the words, in addition to their variations in different linguistics contexts. They tests their new type of embeddings on different NLP tasks including NER. They used the CoNLL 2003 data-set and they built a baseline using character-based representation, pre-trained word embeddings, two BiLSTM layers and CRF layer. The baseline got an F1-score of 90.15%, adding ELMo to it enhanced the model and got an F1-score of 92.22%.

The use of ELMo embeddings improved the performance of most NLP tasks. However, it is not easy to integrate it into neural network architectures and there are several ways to do it, e.g., weighting the three layers or using only the first or last one. Also, it uses a task specific architecture that considers the embeddings as additional features [37]. Another new type of embeddings was presented in [19], it is called Bidirectional Encoder Representations from Transformers (BERT). It uses unlabelled text by jointly conditioning on left and right context in all layers to train bidirectional word representations. They compared several approaches using BERT on CoNLL 2003 data-set for NER and the best one got an F1-score of 92.8%.

In [4], they used their new proposed type of embeddings called Contextual string embeddings for the NER task on CoNLL 2003 English and German data. The embeddings combine the advantages of all other contextual embeddings which are training on large unlableled data-set, taking the context into consideration and model the words as sequence of characters to better handle misspelled words and prefixes and suffixes of words. They achieved an F1-score of 93.09% and 88.33% for English and German languages respectively.

[3] proposed their own Pooling contextualized embeddings for NER task. They used the open source FLAIR framework [1] to build the NER system, it is implemented using BiLSTM-CRF sequence labeling. The system aggregates contextual embeddings of all unique strings by a pooling technique. They used CoNLL 2003 data-set and WNUT-17 task [18] to evaluate their model. The system achieved high results of 93.18% F1-score on CoNLL 2003 English data, 88.27% F1-score on CoNLL 2003 German, 90.44% on CoNLL 2003 Dutch and 49.59% F1-score on WNUT-17. The results achieved on the CoNLL 2003 data are considered the state-of-the-art results in the NER task for these languages.

Code-Mixing

There are several successful attempts to collect CM data for different languages and different purposes. For instance, [45] created a new English-isiZulu code-switched speech corpus for automatic speech recognition. [22] presented a code-switched Arabic-English text corpus that is collected from the web. They used the corpus to build a language model for Arabic-English CS. Also, [15] created a romanized code-switched Algerian Arabic-French corpus collected from Algerian newspaper website and annotated it with word-level language id.

In [40] they created their own code-switched Modern Standard Arabic and Moroccan Arabic (MSA-MA) data-set to use it for language identification. They collected the data from Moroccan internet discussion boards for varying subjects and blogs.

Also, [13] implemented a RNN system to detect the language of code-switch data such as English-Spanish, English-Nepali, Mandarin-English and Modern Standard Arabic-Egyptian Arabic. They used the Twitter data provided by the EMNLP Code-switching Workshop [43].

Several other works have been conducted for code-switching identification for Egyptian Arabic and MSA data such as [5] that used CRF classifier and [39] that used a RNN model.

Concerning the NER on code-mixing data, [21] introduced a hybrid approach for NER from CS English-Hindi and English-Tamil. They used a classifier based on CRF and they achieved F-score of 62.17% for English-Hindi and 44.12% for English-Tamil. In [6] proposed a Bengali-English code-mixed data-set in the domain of sports and tourism for NER task. They also compared four machine learning approaches for NER on code-mixed. The best performance they achieved was 92.31% F1-score for sports data using CRF and 70.63% F1-score using SVM for the tourism data.

[1] https://github.com/zalandoresearch/flair

Lately, in FIRE'2015 a shared task was established to collect and recognize entities from CM social media data for Hindi, Malayalam, Tamil and English languages [36]. Furthermore, in CALCS 2018 shared task [1] they concentrated on NER for CS from social media data for English-Spanish (ENG-SPA) and Modern Standard Arabic-Egyptian dialect (MSA-EGY). The best performance achieved for ENG-SPA was 63.76% F1-score using BiLSTM, character- and word-based representation. About the MSA-EGY the best result was 71.62% F1-score using BiLSTM-CRF along with an embedding layer.

[42] proposed an NER tool for Hindi-English code-mixed data. They implemented two different models one using CRF classifier and another one using LSTM model composed of two bidirectional layers. The performance of their models was 72.06% and 64.64% F1-score respectively.

Data Collection and Annotation

We present our code-switched corpus which is an extension to our work in [38]. The corpus is composed of 6,525 sentences that contains 136,574 tokens. It has 22,705 (16.6%) English words and 113,869 (83.4%) Arabic words. The data was gathered through three different sources. The first one is collecting 2,303 Egyptian-English CM sentences from Twitter. The second one is taking 1,150 sentences from the transcribed speech corpus for conversational Egyptian Arabic [23]. The last one is translating 3,072 sentences from the Arabic ANERCorp and AQMAR data-sets for Named Entity Recognition.

After collecting the data, the annotation of the entities started by identifying the boundaries of the named entity and then assigning the correct NE type. Our annotations followed the Named Entity Annotation guidelines for the shared task of CoNLL-2003 [44]. It is concerned with four types of entities: persons (PER), locations (LOC), organizations (ORG) and names of miscellaneous (MISC), that do not belong to any of the three types. Words that are not named entities are tagged with O. Table 1 shows the number of sentences containing entities in each data-set and their percentages. The total number of sentences in the corpus containing entities is 4,948 sentences, which is 75.8% from the total number of sentences.

Table 1. Number of sentences containing entities in each data-set

Data-set	No. of Sentences
Twitter	1,477 (64.1%)
Translated	3,059 (99.6%)
Transcribed Speech	412 (35.8%)
Total	4,948 (75.8%)

The total number of NEs in the corpus is 17,577 tokens. Table 2 shows the total number of words under each NE class. The person class contains the highest

number of entities which is 6,534 entities and the organization class contains the minimum number of entities which is 3,100 entities.

Table 2. Number of entities in each entity type in the final data-set

Entity Type	Words	% of Total Words
Person	6,534	4.8
Location	4,219	3.1
Organization	3,100	2.3
Miscellaneous	3,724	2.7
Total entities	17,577	12.9

Twitter Data-set

The first part of the corpus is composed of data harvested from Twitter by querying the Twitter API[2]. We implemented different approaches to collect tweets containing CM data. In the first phase of collection, one of the approaches was randomly selecting some tweets by gathering them using a query requiring the tweets to have a hash tag. A big percentage of the hash tags are written in the English language. The tokenization of the hash tags is automatically done while collecting the tweets. Moreover, the query required that the tweets should contain Arabic language. Another filtering criteria was done by checking that the tweets contain at least two English words to guarantee that the sentence contains CM text. However, words such as http or via were not counted.

In the second phase of collection, other techniques were used to search for tweets that contain named entities. We used some of the named entities found in ANERCorp data set as keywords in the search queries to collect more tweets containing NEs. Moreover, in order to guarantee having sentences containing entities, we created a list of names of famous persons and used them as seeds to collect tweets.

The total number of collected sentences using Twitter is 2,303 sentences which contains 38,281 tokens. As shown in Table 3, this part of the corpus has 5,810 entities which is 15.2% from the total number of words. It consists of 7,405 (19.3%) English words and 30,876 (80.7%) Arabic words. We can observe that the miscellaneous class contains the highest number of entities and the location class contains the lowest number of entities.

Translated Data-Set

The second part is gathered by translating some of the existing annotated Arabic NER data. In the first collection phase, the selected sentences were randomly chosen from the Arabic ANERCorp data-set. In some sentences, we translated

[2] https://developer.twitter.com/en/docs/api-reference-index

Table 3. Number of entities in each entity type in the Twitter data-set

Entity Type	Words	% of Total Words
Person	1,907	5.0
Location	550	1.4
Organization	1,098	2.9
Miscellaneous	2,255	5.9
Total entities	5,810	15.2

all the entities they contained. Other sentences we translated one or two entities only, in order to have Arabic entities in addition to the English entities in this part of the corpus. During the second phase of collection more sentences were chosen from the same data-set in addition, to sentences selected from AQmar data-set.

This technique was time efficient as there was no need to annotate the data. However, sometimes it did not generate correct translations for miscellaneous or organization entities as they might have several meanings. Thus, a manual check was done on data to ensure that the words were translated correctly.

The number of translated sentences is 3,072 sentences which contains 78,215 tokens. They are composed of 8,549 (10.9%) English words and 69,666 (89.1%) Arabic words. As shown in Table 4, the total number of named entities it has is 11,391 which is 14.6% of the total number of words. The person class contains the highest number of words which is equal to 4,516 words, and the miscellaneous class contains the minimum number of words which is equal to 1,348 words.

Table 4. Number of entities in each entity type in the translated data-set

Entity Type	Words	% of Total Words
Person	4,516	5.8
Location	3,601	4.6
Organization	1,926	2.5
Miscellaneous	1,348	1.7
Total entities	11,391	14.6

Transcribed Speech Data-set

The third part of our corpus is composed of data from the transcribed speech data-set [23]. They gathered spontaneous speech through informal interviews. The interviews topics were technical ones to have a higher probability to contain CM. They manually transcribed the corpus and they formed a total of 1,234 sentences and 17,769 tokens. In the original speech corpus the sentences were divided into 124 monolingual Arabic, 125 monolingual English and 985 mixed.

Overall, the data-set contains 79.8% of code-mixing, 10.1% of English and 10% of purely Arabic.

Table 5. Number of entities in each entity type in the transcribed speech data-set

Entity Type	Tokens	% of Total Words
Person	111	0.6
Location	68	0.3
Organization	76	0.4
Miscellaneous	121	0.6
Total entities	376	1.9

As shown in Table 5, this part of the corpus contains 1,150 sentences containing 20,078 tokens. As some pre-processings were performed on the data and added more tokens/sentences. It has 13,327 (66.4%) Arabic and 6,751 (33.6%) English words. The miscellaneous class contains the maximum number of entities which is equal to 121 words and the location is the lowest which is equal to 68 words only.

Pre-trained Embeddings

Word embedding or representation is done by mapping every word in a sentence to vector in multi-dimensional space. This leads to having semantically similar words grouped close to each other in the space. As learning independent representations for the types of words from the training data alone is difficult. Thus, Adding word embedding to deep learning models enhances the performance by specifying syntactic and semantic word relationships. There are two main categories of pre-trained word embeddings: Classical and Contextual word embeddings. We investigated using several types of embeddings from these categorizes along with our DNN model architecture for NER on CM data.

Classical Word Embeddings

As stated before the classical word embeddings does not take into consideration the context of the words. We used four different types of classical word embeddings, two for Arabic, one for English language and one for Arabic-English CM data. The first one is our pre-trained word embeddings model for Arabic languages. We used the Word2vec (W2V) algorithm to generate and save our own Arabic word embedding model. The W2V algorithm is based on deep learning in addition to the use of skip-gram and continuous Bag-of-Words (CBOW) [29]. The model of W2V was trained using an independent Arabic news-wire data-set.

The second one is FastText embeddings, it is created by Facebook as an extension of W2V. They released several pre-trained models trained on Wikipedia for different languages and we used the Arabic one that generates vectors with

dimension equal to 300. The main advantage of FastText compared to W2V, it can generate embeddings for unseen words from their character N-gram features; words that are not processed during training [10].

Regarding the English word embedding model, we used GloVe for obtaining vector representations for words. The pre-trained model of GloVe we used was trained on Wikipedia 2014 and Gigaword 5 data [33].

Furthermore, we used the pre-trained Bilingual CS Embeddings (Bi-CS), it is a bilingual Egyptian Arabic-English word embeddings from [24]. The authors trained several word embeddings using multiple algorithms that rely on different levels of cross-lingual supervision. We used the Bi-CS embeddings as it showed most promising performance. They have two different Bi-CS models that we tried one that is trained using Skip-gram (Bi-CS-skip) and the other one using CBOW (Bi-CS-cbow).

Out-of-Vocab (OOV) words that were not found in W2V, GloVe or Bi-CS models are being represented by a vector of zeros in the embedding layer.

Contextual Word Embeddings

It is challenging to learn high quality representations, because of the different characteristics of the words and how they change based on the linguistic contexts [35].

Thus, contextual word embeddings are considered to be the new approach of representing the vector of a word in a given text compared to word2Vec and GloVe model. It takes into consideration the context of a word in given sentence. In other words, they provide different vector representation of single word. We used the following types of contextual embeddings and compared their performances in our models: ELMo, BERT, Contextual string embeddings, and Pooled FLAIR embeddings.

The first type of embeddings is ELMo, we used the Arabic ELMo representations implemented by [35]. It combines all layers of a deep pre-trained neural network. Also, they are character based which means they can generate embeddings for out-of-vocabulary words that did not occur in the training data.

The second type is BERT, it is a state-of-the art in eleven NLP tasks including NER [19]. The main BERT model that we used during the training was BERT Multilingual model which contains 104 different languages including the Arabic and English languages.

Contextual string embeddings is the third type of embeddings we investigated, they are a new type of embeddings implemented by [4]. Beside taking the context into consideration they are trained without explicit notion of words which led to modelling the words as sequence of characters.

The last one as far as we know is the newest, it is the Pooled FLAIR embeddings. It solves the disadvantage of the contextual string embeddings which is generating meaningful embeddings of rare strings used in under-specified context. It aggregates the contextualized embeddings of all unique strings it finds. Then, it refines one word embedding for all contextualized instances [3].

BiLSTM-CRF

RNN learns long distance dependencies because of maintaining a memory based history information. However, practically, they fail due to the vanishing or exploding gradient [28]. In order to solve this vanishing problem Long-Short-Term-Memory (LSTM) were designed as a variant to RNN [25]. LSTM network contains connected memory blocks/cells instead of the traditional nodes of the RNN. However, the disadvantage of using LSTM, is not being able to check the future context as it checks the previous one only.

For the purpose of taking benefit of the previous and future contexts, BiLSTM was proposed as an extension to LSTM. It represents each sequence forward and backward as two separate hidden states to save the previous and future information. At the end the two hidden states are concatenated to form the output [41]. For the sake of predicting the current tags, there are two common ways

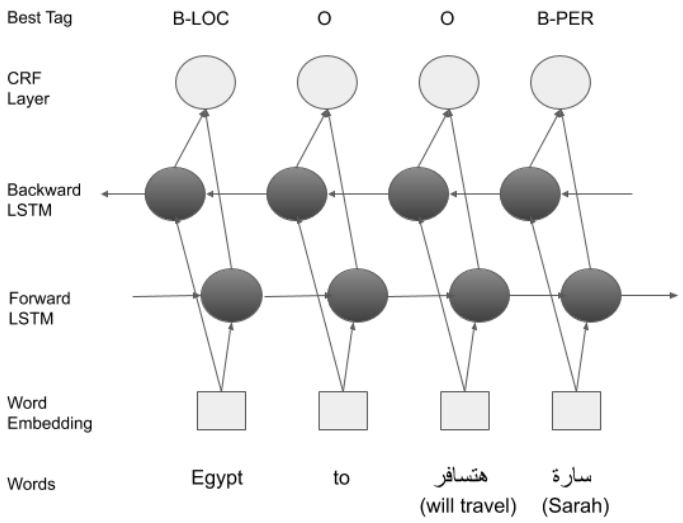

Fig. 1. Our main BiLSTM-CRF model Architecture. The example illustrates the input sentence "Sarah will travel to Egypt" and its output predicted tags.

to make use of the previous and future tag information. One way is to predict a distribution of tags step by step then use beam-like decoding to find the best sequence of tags this could be achieved by using Maximum Entropy Markov model. The other way is to use CRF model. It focuses on sentence level instead of individual positions [26].

In order to combine the advantages of BiLSTM and CRF networks, we constructed all our models using BiLSTM-CRF. The model architecture is composed of three layers as shown in Figure 1. The first one is the input layer. It contains

the word embeddings, we tried several types of word embeddings as will be explained later. The second one is the hidden layer of the BiLSTM. The last one is the output layer. It is where the CRF layer calculates the probability distribution over all labels of the previous and future tags to predict the current best tag. As it has been proven that using character-level embeddings is useful for morphological rich languages and they can handle the out-of-vocabulary words [27]. Thus, we constructed a second model architecture contains four layer. The first one is the word embeddings layer. The second one is the character embeddings layer. Then, a concatenation of the first and second layers is done to be given as input for the third layer which is the BiLSTM layer. The final one is the CRF layer.

Experiments and Results

Our experiments are divided into two parts, training the models with and without CM data. However, we unified in all our following experiments the testing data set which contains 1,219 sentences and 28,581 tokens.

Multiple Monolingual Data-sets

We implemented our baseline system and enhanced it using the pooling technique. We did not use in the baseline or pooling systems any CM data in the training process, it was just combining multiple monolingual data as will be explained in details later.

Baseline In order to build the baseline, two models were created. The first one is the Arabic model, the Arabic data we selected for training is the ANERCorp and AQMAR. It contains 225,000 words annotated for the NER task. Concerning the testing data in all systems, we used our collected CM Arabic-English corpus that consists of 28,581 annotated words. The performance of the Arabic model is 27.7% F1-score. The second one is the English model, the data we selected for training is CoNLL 2003, it contains 206,931 annotated words. The performance of the English model is 7.9% F1-score. Both systems achieved low F1-score, which is expected as the training data contains only one of the two languages and the testing data contains mixed sentences.

The baseline starts by detecting the language of each word in the testing data-set. Based on the detected language the predicted tag is taken from either the Arabic or English predictions. The overall F1-score achieved by the baseline system is 52%, which is still a low expected performance as the predictions do not consider code-mixing context.

Data Pooling We introduced the concept of data pooling to overcome the problem of lack of large CM data. Pooling is done by combining annotated Arabic and English data-sets together to form the training data-set. We used the same Arabic and English data-sets as the baseline but we combined them together, thus, the total number of words in the training file is 431,931 tokens. Concerning the testing data we used our collected CM Arabic-English corpus (same as the baseline). The training is performed using Nadam optimization function [20], softplus activation function [32], batch size of 32 and epochs number of 10.

As the corpus contains two different languages, we suggested to load two classical word embedding models to cover English and Arabic words. For the Arabic we used our W2V model and for English we used the GloVe model. As a result of using the pooling technique by combining the two training data-sets and using the collected CM corpus for testing, the best performance achieved is 60% F1-score. This result is expected taking into consideration that the training and testing data belongs to totally different context due to the limited amount of CM data at our previous work.

Code-Mixed Data-set

As an extension of the previous work, we collected more data and were able to train and test with CM data. The total number of sentences in the training file is 5,306 sentences which contains 107,993 tokens. We explored the combination of BiLSTM-CRF architecture with different types of embeddings. In addition, we used the open-source FLAIR framework to try the same architecture of BiLSTM-CRF with their proposed different embeddings in order to identify the best performance for NER on our CM data-set.

FLAIR is a recent NLP framework designed to train and distribute text classification, language models and sequence labelling. In addition, it unifies the use of many different word embeddings as well as random combinations of embeddings [2]. The FLAIR system was trained several times using our CM training data-set each time with a different type of embeddings and we took advantage of its feature of combining two types of embeddings.

The setups of the different models we tried were as follows:

BiLSTM-CRF : We explored using different types of embeddings with BiLSTM-CRF model architecture. The first type is the same one we used in the pooling technique but here we used it with the newly collected CM training data. It is composed of the two pre-trained word embeddings GloVe for English and W2V for Arabic. The second one is using the Bi-CS word embeddings with its two models of Bi-CS-skip and Bi-CS-cbow to see the effect of using CS word embeddings on the performance. The last two are using BERT Multilingual and ELMo Arabic embeddings.

FLAIR (BiLSTM-CRF) : We investigated using FLAIR and the BiLSTM-CRF architecture with the following different types of embeddings explained before: BERT, ELMo, FastText, Bi-CS-skip, Bi-CS-cbow, Contextual string embeddings and Pooled embeddings. In addition, we tried combining the highest performing embedding type with other embeddings.

We tuned the hyper-parameters of the models using grid search, the dropout ranged from 0.2 to 0.8, batch size ranged from 10 to 32, dense layer size ranged from 50 to 1024, BiLSTM size ranged from 100 to 1024 and number of epochs ranged from 50 to 150. Concerning the optimization function we used Adam optimizers and stochastic gradient descent (SGD) optimizers with learning rate of 0.001 to 0.1. Regarding the activation function we used the tanh function.

As shown in Table 6, the results of BiLSTM-CRF with BERT embeddings is equal to 53.73% F1-score which is considered the lowest performance. This could be due to the type of data BERT multilingual model used in training to

Table 6. Results of the NER models with different types of embeddings

Model	Precision	Recall	F1-Score
BiLSTM-CRF			
+BERT	63.05	46.81	53.73
+W2V & GloVe	70.53	62.09	66.04
+ELMo	70.00	69.66	69.83
+FastText	76.55	67.80	71.91
+Bi-CS-cbow	78.05	67.57	72.43
+Bi-CS-skip	78.08	67.71	**72.53**
FLAIR (BiLSTM-CRF)			
+Bi-CS-cbow	63.58	51.80	57.09
+Bi-CS-skip	66.48	52.52	58.68
+ELMo	79.34	61.47	69.27
+BERT	82.40	60.90	70.04
+FastText	75.65	66.00	70.49
+Contextual string embeddings	74.58	70.42	72.44
+Pooled embeddings	76.83	72.66	74.69
+Pooled embeddings & BERT	81.29	63.76	71.47
+Pooled embeddings & ELMo	77.00	69.38	72.99
+Pooled embeddings & Bi-CS-skip	77.93	73.85	75.84
+Pooled embeddings & FastText	79.15	76.28	**77.69**

generate the pre-trained embeddings which is Wikipedia pages. The language used in Wikipedia is a MSA which is different than the one used in the CM training data. In addition, the embeddings does not contain CM sentences.

The same model of BiLSTM-CRF and the two pre-trained word embeddings for English and Arabic that we previously trained with the pooling technique got higher F1-score by 6.04% using CM training data. However, it is still low F1-score compared to other models and this could be due to having two different embedding models for English and Arabic that give to the same word in both languages different embeddings. Adding ELMo or FastText embeddings in our model enhanced the performance and got an F1-score of 69.83% and 71.91% respectively. In addition, when we used Bi-CS-cbow embeddings the model achieved an F1-score of 72.43%. However, using the Bi-CS-skip embeddings outperformed the results of the other embeddings used with our model and got an F1-score that is equal to 72.53%.

The embeddings that got the lowest results equal to 57.09% and 58.68% F1-score with FLAIR system are the Bi-CS-cbow and Bi-CS-skip respectively. Although, this type of embeddings got the highest results compared to other embeddings with the normal BiLSTM-CRF model.

There is no big difference in the performace of FLAIR system with ELMo, BERT or Arabic FastText embeddings, they achieved an F1-score equal to 69.27%, 70.04% and 70.49% respectively.

The performance is improved more while using FLAIR with Contextual string embeddings and Pooled embeddings as they are the newest types of embeddings and they deal efficiently with unseen words. They achieved an F1-score of 72.44% and 74.69% respectively.

In order to take advantage of FLAIR feature of combining several embeddings together, we evaluated combining the Pooling embeddings that achieved the highest F1-score with the other existing types. Adding BERT and ELMo to it did not achieve higher results than the Pooling alone and they got an F1-score of 71.47% and 72.99% respectively. However, combining it with Bi-CS-skip enhanced the F1-score with 1.15%. In addition, the Pooling with FastText embeddings performs particularly well on our task of NER on CM data. It produced the highest results among all other models which is equal to 77.69%. We selected the two models with the highest F1-score to check the effect of adding character-level embeddings on them. Thus, we tried first the FLAIR and the BiLSTM-CRF models with character-level embeddings alone. Then, we added character-level embeddings to FLAIR with Pooled embeddings & FastText and to BiLSTM-CRF with Bi-CS-skip.

Table 7. Results of the highest models with character-level embeddings

Model	Precision	Recall	F1-Score
FLAIR (BiLSTM-CRF)			
+Character Embeddings	71.16	64.04	67.41
+Character Embeddings & Pooled embeddings & FastText	79.28	75.61	**77.40**
BiLSTM-CRF			
+Character Embeddings	77.53	72.33	74.84
+Character Embeddings & Bi-CS-skip	77.30	73.47	75.34

We observed that having character embeddings implemented by FLAIR alone got low results of 67.41% F1-score. Having character embeddings with our BiLSTM-CRF model got a F1-score of 74.84%. In addition, adding character embeddings along with the Bi-CS-skip embeddings enhanced the F1-score with 2.81% as shown in Table 7. Nevertheless, it decreased the performance by 0.26% while being added to the FLAIR model with the Pooled embeddings & FastText. This means that character-level embeddings does not add much benefit to the new type of contextual embeddings along with the classical word embeddings in our case. Thus, at the end the model of FLAIR with Pooled embeddings & FastText remains the one with highest performance that is equal to 77.69% and it outperforms the previous results of the baseline by 25.69% absolute F1-score.

To further investigate the result of the model with highest F1-score, the following Table 8 shows the results for each entity type. The entity type person had the highest F1-score which is equal to 89.88%, which is expected as in our corpus the maximum number of entities belongs to the person class. Followed by the entity types location and organization which are equal to 84.52 % and

61.66% F1-score respectively. The minimum F1-score is 37.07% that belongs to the miscellaneous class.

Table 8. Detailed results of each entity type for the highest model

Entity Type	Precision	Recall	F1-Score
Person	86.65	93.36	89.88
Location	82.02	87.17	84.52
Organization	75.33	52.19	61.66
Miscellaneous	41.53	33.48	37.07

Conclusion & Future Work

Code mixing became a common studied linguistic behaviour and a huge amount of CM data is generated through the different social media platforms. It is more difficult to apply traditional NER techniques on such data specially Arabic-English CM text, due to the lack of large annotated corpus and the informality of the text. In this paper, we have presented the first annotated Arabic-English CM corpus for NER. The corpus contains 6,525 sentences. Along with different deep learning models for NER on CM Arabic-English data. The models are composed of BiLSTM-CRF network and pre-trained word embeddings models. A baseline was built by training two different models: one for Arabic and another one for English, for the sake of detecting the language of the testing words and getting the predicted tag accordingly. The performance of the baseline was 52% F1-score. First to improve the results, we introduced a data pooling approach by combining different training data-sets for English and Arabic. As at the beginning the size of the CM corpus was very limited, it was only kept for testing and evaluation. Moreover, the model used two different pre-trained word embedding models one for Arabic and another one for English.

In addition, we have combined with our model architecture different types of classical and contextual pre-trained word embedding models to compare their performance and achieve higher results. The usage of Pooled embeddings & Fast-Text as pre-trained word embeddings in the model achieved highest performance that is equal to 77.69% F1-score.

For future work, we may use some transfer learning techniques to import models trained on other CM languages. In addition, we can train our own contextual CM word embeddings and add it to the BiLSTM-CRF model. Moreover, we can try fine tuning BERT model and training it with CM data.

References

1. Aguilar, G., AlGhamdi, F., Soto, V., Diab, M., Hirschberg, J., Solorio, T.: Named entity recognition on code-switched data: Overview of the calcs 2018 shared task. In: Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching. pp. 138–147 (2018)
2. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: Flair: An easy-to-use framework for state-of-the-art nlp. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 54–59 (2019)
3. Akbik, A., Bergmann, T., Vollgraf, R.: Pooled contextualized embeddings for named entity recognition. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 724–728 (2019)
4. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1638–1649 (2018)
5. Al-Badrashiny, M., Elfardy, H., Diab, M.: Aida2: A hybrid approach for token and sentence level dialect identification in arabic. In: Proceedings of the Nineteenth Conference on Computational Natural Language Learning. pp. 42–51 (2015)
6. Banerjee, S., Naskar, S.K., Rosso, P., Bandyopadhyay, S.: Named entity recognition on code-mixed cross-script social media content. Computación y Sistemas **21**(4), 681–692 (2017)
7. 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. pp. 284–293. Association for Computational Linguistics (2008)
8. Benajiba, Y., Rosso, P., Benedíruiz, J.M.: Anersys: An arabic named entity recognition system based on maximum entropy. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 143–153. Springer (2007)
9. Benikova, D., Biemann, C., Kisselew, M., Pado, S.: Germeval 2014 named entity recognition shared task: companion paper (2014)
10. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
11. Bokamba, E.G.: Are there syntactic constraints on code-mixing? World Englishes **8**(3), 277–292 (1989)
12. Caroline Sabty, M.E., Abdennadher, S.: Arabic named entity recognition using clustered word embedding (2018)
13. Chang, J.C., Lin, C.C.: Recurrent-neural-network for language detection on twitter code-switching corpus. arXiv preprint arXiv:1412.4314 (2014)
14. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research **12**(Aug), 2493–2537 (2011)
15. Cotterell, R., Renduchintala, A., Saphra, N., Callison-Burch, C.: An algerian arabic-french code-switched corpus. In: Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme. p. 34 (2014)
16. David Awad, Caroline Sabty, M.E., Abdennadher, S.: Arabic name entity recognition using deep learning (2018)
17. Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., Bontcheva, K.: Analysis of named entity recognition and linking for tweets. Information Processing & Management **51**(2), 32–49 (2015)

18. Derczynski, L., Nichols, E., van Erp, M., Limsopatham, N.: 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)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
20. Dozat, T.: Incorporating nesterov momentum into adam (2016)
21. Gupta, D., Tripathi, S., Ekbal, A., Bhattacharyya, P.: A hybrid approach for entity extraction in code-mixed social media data. MONEY **25**, 66 (2016)
22. Hamed, I., Elmahdy, M., Abdennadher, S.: Building a first language model for code-switch arabic-english. Procedia Computer Science **117**, 208–216 (2017)
23. Hamed, I., Elmahdy, M., Abdennadher, S.: Collection and analysis of code-switch egyptian arabic-english speech corpus. In: LREC (2018)
24. Hamed, I., Zhu, M., Elmahdy, M., Abdennadher, S., Vu, N.T.: Code-switching language modeling with bilingual word embeddings: A case study for egyptian arabic-english (2019)
25. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
26. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
27. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
28. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
29. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
31. Mohit, B., Schneider, N., Bhowmick, R., Oflazer, K., Smith, N.A.: Recall-oriented learning of named entities in arabic wikipedia. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 162–173. Association for Computational Linguistics (2012)
32. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
33. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
34. Peters, M.E., Ammar, W., Bhagavatula, C., Power, R.: Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108 (2017)
35. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
36. Rao, P.R., Malarkodi, C., Ram, R.V.S., Devi, S.L.: Esm-il: Entity extraction from social media text for indian languages@ fire 2015-an overview. In: FIRE Workshops. pp. 74–80 (2015)
37. Reimers, N., Gurevych, I.: Alternative weighting schemes for elmo embeddings. arXiv preprint arXiv:1904.02954 (2019)

38. Sabty, C., Elmahdy, M., Abdennadher, S.: Named entity recognition on arabic-english code-mixed data. In: 2019 IEEE 13th International Conference on Semantic Computing (ICSC). pp. 93–97. IEEE (2019)
39. Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., Solorio, T.: Multilingual code-switching identification via lstm recurrent neural networks. In: Proceedings of the Second Workshop on Computational Approaches to Code Switching. pp. 50–59 (2016)
40. Samih, Y., Maier, W.: Detecting code-switching in moroccan arabic social media. SocialNLP@ IJCAI-2016, New York (2016)
41. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
42. Singh, K., Sen, I., Kumaraguru, P.: Language identification and named entity recognition in hinglish code mixed tweets. In: Proceedings of ACL 2018, Student Research Workshop. pp. 52–58 (2018)
43. Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A., et al.: Overview for the first shared task on language identification in code-switched data. In: Proceedings of the First Workshop on Computational Approaches to Code Switching. pp. 62–72 (2014)
44. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. pp. 142–147. Association for Computational Linguistics (2003)
45. van der Westhuizen, E., Niesler, T.: Automatic speech recognition of english-isizulu code-switched speech from south african soap operas. Procedia Computer Science **81**, 121–127 (2016)
46. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2145–2158 (2018)
47. Yan, S., Hardmeier, C., Nivre, J.: Multilingual named entity recognition using hybrid neural networks. In: The Sixth Swedish Language Technology Conference (SLTC) (2016)