IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 15, 2020, accepted February 1, 2020, date of publication February 11, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973319

Transfer Learning for Arabic Named Entity Recognition With Deep Neural Networks

MOHAMMAD AL-SMADI[1], SAAD AL-ZBOON[1], YASER JARARWEH[1,2], AND PATRICK JUOLA[2]

[1]Computer Science Department, Jordan University of Science and Technology, Irbid 22110, Jordan
[2]Mathematics and Computer Science Department, Duquesne University, Pittsburgh, PA 15282, USA

Corresponding author: Mohammad Al-Smadi (masmadi@just.edu.jo)

This work was supported by the Jordan University of Science and Technology under Grant 20170107.

ABSTRACT The vast amount of unstructured data spread on a daily basis rises the need for developing effective information retrieval and extraction methods. Named Entity Recognition is a challenging classification task for structuring data into pre-defined labels, and is even more complicated when being applied on the Arabic language due to its special traits and complex nature. This article presents a novel Deep Learning approach for Standard Arabic Named Entity Recognition that proved its out-performance when being compared to previous works. The main aim of building a new model is to provide better fine-grained results for use in the Natural Language Processing fields. In our proposed methodology we utilized transfer learning with deep neural networks to build a Pooled-GRU model combined with the Multilingual Universal Sentence Encoder. Our proposed model scored about 17% enhancement when being compared to previous work.

INDEX TERMS Natural language processing, deep learning, transfer learning, ANER, universal sentence encoder, Bi-LSTM.

I. INTRODUCTION

In the 1990s, Named Entity Recognition (NER) was introduced for the first time as an information extraction task at the Message Understanding Conferences [1], and since that date, it had a great attention in the research community. The importance of NER in the last few years has taken huge concern due to its importance in the growing field of Natural Language Processing (NLP); for being the first step of semantic labeling [2]. NER's main aim is to label or classify entities in a particular text based on pre-defined labels or tags (e.g. person, location, or organization, etc.) [3].

A lot of work has been done in English and other widely spread languages, such as: Spanish and Chinese with high accuracy records. Researchers depend on three main approaches when building NER systems, these approaches as described in [1] as: linguistic-rule-based, machine-learning-based, and hybrid-based approaches.

The Linguistic approach uses rule-based models that are manually written by linguists. In this approach, a set of rules or patterns is being defined in order to distinguish the NEs in a text [4]. In 1995, [5] developed rule-based NER systems that use specialized dictionaries of names that included countries, names of major cities,..etc. In 1996, [6] also developed a NER rule-based system that uses several conventional entities like persons' names, organizations' names, location names, and so on. The main obstruction of the rule-based techniques is that they require huge grammatical knowledge, in addition to the experience in particular languages. Also, these systems cannot be translated into other languages. This approach was used for years before the emergence of machine learning era; as with the rise of machine learning the first step was the adoption of both approaches into hybrid solution [7], [8].

Machine Learning based techniques use a large amount of annotated training data in order to gain high-level language knowledge. ML models are built based on two types of algorithms: supervised and unsupervised models. Unsupervised NER models do not demand any training data [9]. The main idea of such a model is to create possible annotations from the dataset itself. This learning model is not popular within the ML methods, due to its lack of accuracy with the absence of supervised methods. On the other hand, Supervised models require a huge amount of annotated data to produce a well-trained system. Some examples of the ML techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

being used for NER algorithms cover the artificial neural network (ANN), Hidden Markov Model (HMM), Maximum Entropy Model (MaxEnt), Decision Trees, Support Vector Machine and others [9].

Deep learning is a sub field of ML; which is a combination of multiple processing layers that can learn representations of data in multiple levels of abstraction. There are two basic architectures that are used widely to extract textual representation(character-level or word-level): (i) convolutional neural networks (CNN) based and (ii) recurrent neural networks (RNN) based models [10].

In this research, we are proposing a state-of-the-art approach for Arabic NER based on deep Learning. More precisely, a transfer learning with deep neural network model is implemented for this research purposes. We refer to this model as Pooled-GRU along with Multilingual Universal Sentence Encoder (MUSE) [11]. The proposed model is evaluated using the Arabic NER dataset ($\text{WikiFANE}_{\text{Gold}}$) presented in [12]. Evaluation results are compared to our previous research of using a Bidirectional Long Short-Term Memory (LSTM) with conditional random fields CRF (BI-LSTM-CRF) [3]. Evaluation results are promising and show that the new pooled-GRU model outperforms our previous Bi-LSTM-CRF model with about 17% over the F1 measure.

This article is organized as follows: section II provides a comprehensive survey on NER for English and other languages with a focus on traditional machine learning and deep learning research for Arabic NER. Section III describes the methodology used in named entity recognition models, including the dataset, and the proposed models. Whereas section IV discusses the experimental setup and evaluation along with results. And finally, section VI concludes this research and provides plans for future work.

II. RELATED WORK

A. NER IN ENGLISH AND FOREIGN LANGUAGES

Works to build robust NER systems has been a vital research area since a long time, and many models were tested and built to obtain significant results. HMM specifies a joint probability for a pair of observations and labeled sequences [4]. And thus, the parameters are being trained to make the joint likelihood of training sets as large as possible. Many studies such as [13]–[15] used the HMM on different datasets and different languages. This method has proven its advantages over the rule-based methods, as it can hold more than one feature for each word.

Maximum Entropy (MaxEnt) is a conditional probabilistic sequence model that can declare multi-features of one word, and also it is able to handle long term dependencies [4]. The authors of [16] and [17] Proposed a MaxEnt approach for the task of NER. Their NER system did not use local context within a sentence only, but it also used the repetition of each word inside the same document in order to extract useful features (i.e. global features). In reference [16] the authors have also built a high-performance NER system without the use of disconnected classifiers to handle the global consistency or complex formulation. They used less training data than other systems, but their NER system was able to act as other state-of-the-art NERs. The authors of [18] and [19] evaluated the behavior of the C4.5 algorithm on the task of decision trees learning in recognizing and classifying NEs in text.

SVM is known in the cases of solving multi-class pattern recognition problems. SVM method is well-known because of its good generalization performance and for its applications on pattern recognition problems. The authors of [20] built a NER system that uses only the language-independent features. These features were applied for their NER system on the Indian language. They conducted experiments for finding out the best set of features for NER in Bengali and Hindu languages. They have generated lexical patterns from the context by an unlabeled Bengali news corpus. And then, these patterns were used as features of SVM.

The authors of [21] proposed a mask method by the use of a rich feature set. Their method gained a fine result on different NEs tasks. The experiments and results showed that the proposed method achieved a performance as the state-of-the-art performance in the task of biology named entity extraction.

Reference [22] applied CNN to extract character-level representations of words. Their method used the character representation vector to be combined with the word embedding together before feeding an RNN context encoder (sequence labeling). The authors of [23] utilized a chain of convolutional and highway layers for the aim of generating character-level representations for words. And the final embeddings of words were fed to a bidirectional recursive network.

The authors of [24] suggested a neural re-ranking model for NER, as a convolutional layer along with a fixed-window-size that was used on the top of a character embedding layer. Reference [24] proposed ELMo, which is a word representation that was computed on the top of two-layers of bi-directional language models with character convolutions. The authors of [25] used a bi-directional LSTM in order to extract character-level representations of words.

Likewise reference reference [22], used character-level representation to be combined with pre-trained word-level embedding through a word lookup table. The authors of [26] presented a neural NER model using stack-residual LSTM with a trainable decoder. They extracted the word features from word embeddings as well as character-level RNN.

The authors of [27] enhanced models' ability to be able to handle both cross-lingual and multi-task joint training in a unified way. They utilized a deep bi-directional GRU in order to find out important morphological representations of the character sequence in a word. After that, the character-level representation and word embedding are combined together to be used in the production of the final representation for a word. Recently, reference [28] employs a BERT model with a (CRF) layer to extract NER of the Portuguese language. They also compared the feature-based and fine-tuning based strategies of training.

B. ARABIC NAMED ENTITY RECOGNITION ANER

Arabic is a difficult language that has complex morphological and orthographic behaviors, and this might complicate NER tasks [29]. Despite that, there is a huge amount of researches paying attention to the Arabic named entity recognition (ANER) nowadays. Following is a summary of most cited articles starting from early stages of traditional machine learning ANER methods along to cover the most recent works of deep learning methods.

1) TRADITIONAL AND MACHINE LEARNING METHODS:

The authors of [30] proposed an ANER system based on Hidden Markov Model (HMM). Their model used the stemming process for addressing the inflection and ambiguity of the Arabic language. Their system was fully automated in recognizing NEs, and it was tested using a corpus developed from many sources such as Al Hayat newspaper. An ML system using Decision Trees was proposed in [31]. Their proposed system can extract persons' names, locations, types of crimes, times and date NEs. Their dataset was gathered from online resources, and the obtained F-measure was 81.35%.

The authors of [32] have used the CRF method to replace the Maximum Entropy, with aims to improve their system performance. The features they used in their system were POS tagging, Base Phrase Chunks (BPC), gazetteers and nationalities. They reported results with high accuracy in the general system performance. The indicators were: recall 72.77%, precision 86.90%, and F-measure 79.21%. In reference [33], the authors combined two ML systems for handling Arabic NER, pattern recognition using CRF along with bootstrapping. The features they used included word-level features, POS tagging, BPC, gazetteers, and morphological features. Their system identified various NEs (e.g. Person, Location, Device, Cell Phone, Organization, Car, Date and Time).

The authors of [34] developed a model for the ANER using neural network architecture. The system used two methods to extract 4 types of named entity: person, organization, location, and miscellaneous. The conducted experiment were made to compare between the Decision Tree (DT) and Neural Network applied on the same data. Evaluation results showed that the neural networks achieved 92% of precision, whereas the DT had 87% of precision.

The authors of [35] built multiple classifiers of SVM combined with the CRF approach. They used ACE datasets in their evaluation process. Their results showed that it cannot be stated if CRF is better than SVM or not in ANER or vice versa. They stated that each NE type was sensitive to different features, and thus, each feature has a different role in recognizing the NE with different degrees.

The authors of [36] investigated the impact of word representations on the ANER systems. They presented many approaches used in the integration of word representations with NER. Also, they provided a comparison of commonly used neural word embedding algorithms. The dataset used in the evaluation was AQMAR dataset, the experiments showed that word representation feature increased significantly for the supervised ANER system. At the end, they concluded that the performance was improved when combining different approaches together.

The authors of [34] used 90% of their dataset as training set, while keeping the remaining 10% for testing. The training set represented the input values for the classification model of the Artificial Neural Network (ANN). Their performed experiment was about comparing the proposed ANN with the Decision Tree (DT) approach while using the same testing set. The results showed that the ANN approach out-performed the DT in performance and accuracy. Where the ANN achieved 92%, and DT obtained 87% in the precision measurement.

The authors of [37] developed RenA, a NER system to extract NEs from news articles. They built their own corpus for the purpose of RenA evaluation and to be utilized by other researchers in later researches. The evaluation process was to compare the proposed RenA system with another available NER system in the LingPipe toolkit.

2) RECENT DEEP LEARNING METHODS:

The authors of [2] conducted an evaluation criteria that compared two approaches of ANER. The first tested approach was the traditional ML using CRF which was trained via morphological and syntactic features. The second tested approach was a DNN model that used the Bi-LSTM-CRF which was trained via word-level representations. Their results showed that the second DNN model performs better than the CRF model with a 15% enhancement on the F1-score value. Reference [38] used different RNN cells for NER application (e.g. Bi-RNN, Bi-LSTM, and Bi-GRU). They applied their methods to the ANERcorp dataset. Their observation was that bidirectional implementations can achieve maximum accuracy.

The authors of [39] proposed a new method in order to detect and classify ANER. Their approach used deep co-learning, which is based on a semi-supervised learning algorithm. For the training step, they first developed a Wikipedia article classifier by the use of LSTM-DNN, which was used to obtain a semi-labeled dataset for the task of ANER. The evaluation task was conducted on three different ANER datasets, they also compared the results with many state-of-the-art and of-the-shelf ANER approaches. They concluded that their method gave significant results that out-performed the compared approaches applied to the different three datasets.

The authors of [40] experimented with the B-RNN combined with LSTM/GRU for the ANER task, but they did not use any feature engineering or additional preprocessing. They found that DL approaches, particularly the LSTM, are useful in the identification of Arabic NEs and they can expeditiously out-perform other approaches that are based on manually engineered features or rule-based systems. They also concluded that the integration of the pre-trained word embedding can qualify the system to obtain considerable refinements in the recognition task. The quantitative results

TABLE 1. The 8 coarse-grained classes and 50 fine-grained classes with their number of occurrences in the $\text{WikiFANE}_{\text{Gold}}$ dataset.

Coarse-grained: Fine-grained	Count	Coarse-grained: Fine-grained	Count
PER: Athlete	1292	GPE: Continent	673
PER: Politician	6311	GPE: County-or-District	365
PER: Scientist	2144	GPE: GPE-Cluster	628
PER: Businessperson	855	GPE: Nation	6131
PER: Lawyer	110	GPE: Population-Center	6663
PER: Artist	2501	GPE: State-or-Province	2022
PER: Religious-PER	2617	FAC: Airport	216
PER: Police	1297	FAC: Building-Grounds	2034
PER: Group	3536	FAC: Path	747
PER: Engineer	253	FAC: Plant	94
ORG: Government	1744	FAC: Subarea-Facility	159
ORG: Non-Governmental	1982	WEA: Blunt	48
ORG: Educational	1244	WEA: Chemical	7
ORG: Media	1157	WEA: Exploding	31
ORG: Commercial	2166	WEA: Nuclear	71
ORG: Sports	1129	WEA: Projectile	218
ORG: Religious-ORG	193	WEA: Sharp	5
ORG: Entertainment	158	WEA: Shooting	61
ORG: Medical-Science	182	VEH: Air	1443
LOC: Water-Body	2892	VEH: Land	1069
LOC: Land-Region-Natural	946	VEH: Water	3387
LOC: Celestial	1037	PRO: Book	1175
PRO: Food	150	PRO: Drug	105
PRO: Movie	484	PRO: Hardware	848
PRO: Sound	331	PRO: Software	1147
—	—	O	427,654
Total		493,469	

showed excellent improvements in F-score measures, as they gained a high F-score measure of about 88.01% for Bi-LSTM and 87.12% for Bi-GRU.

The authors of [41] studied an NN model of a multi-attention layer to extract ANEs. They used two attention units; the first is the embedding attention layer, and the other one is the self-attention unit. They claimed that their approach improved the performance notably, basically for the unseen words labeling. Their model achieved 91% of the F1 score using the ANERCorpus dataset, which out-performes the existing approaches in a notable margin.

The authors of [42] proposed CasANER system that can recognize and annotates ANEs. The system contains two types of transducer cascades, which are the analysis and the synthesis which were implemented using the CasSys tool. To go with CasANER, they made a detailed-deep ANE categorization to create a category hierarchy. This hierarchy depends on representative Arabic Wikipedia corpus which contains articles that were extracted from diverse Arabic countries. Their evaluation measurement values showed that CasANER proved its reliability as an impact of its encouraging results.

The authors of [43] proposed a character-level tagger using a deep bi-directional LSTM architecture to extract the ANER. They used the characters as primary representation. Their work showed that using the character level gives good performance over multi-languages, and that was without the need for hand-engineered features or specific language from external resources.

TABLE 2. Size of training, validation, and testing datasets.

Dataset	WikiFANE-Gold
# OF Training Tokens	300k
# OF Validation Tokens	100k
# OF Test Tokens	100k

III. RESEARCH METHODOLOGY

In this research, we proposed two different deep learning models for ANER: 1) baseline Bi-LSTM-CRF following our work in [3]. 2) Pooled-GRU with Multilingual Universal Sentence Encoder (USE) [11]. Both models are evaluated using the $\text{WikiFANE}_{\text{Gold}}$ dataset [12]. The following Sub-sections discuss the methodology in more detail.

A. DATASET

As presented in **Table 1**, the $\text{WikiFANE}_{\text{Gold}}$ consists of 8 coarse-grained classes spans over 50 fine-grained classes. The 8 coarse-grained classes are: **(1) Person (PER)**, **(2) Location (LOC)**, **(3) Organization (ORG)**, **(4) Geopolitical (GPE)**, **(5) Facility (FAC)**, **(6) Vehicle (VEH)**, **(7) Weapon (WEA)**, and **(8) Product (PRO)**. **Figure 1** depicts the distribution of ANER coarse-grained classes that used to train and evaluate our proposed models.

In total, the dataset consists of nearly 500k tokens. 300k tokens were used as a training dataset, and 100k tokens were used as each validation and test datasets (see **Table 2**).

B. BASELINE MODEL

The baseline model is based on our previous research Bi-LSTM-CRF [3]. As depicted in Figure 2, BI-LSTM-CRF

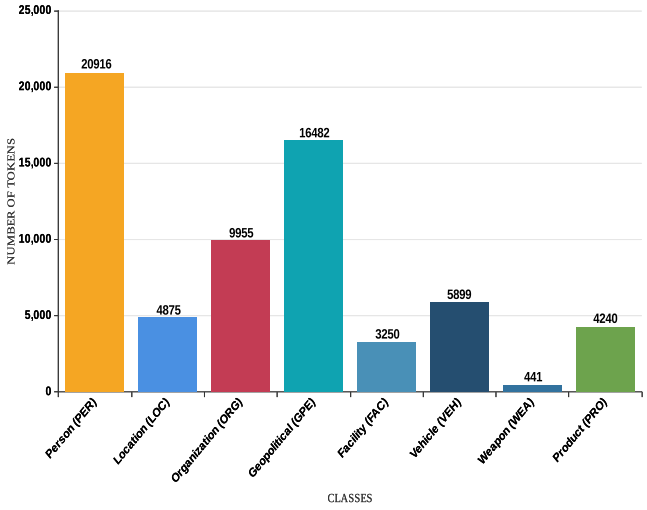

FIGURE 1. The 8 coarse-grained classes distribution based on number of occurrences in the dataset.

is implementing a Bidirectional Long Short-Term Memory (Bi-LSTM) model followed by a conditional random filed classifier (CRF) to learn Arabic named entities. The Bi-LSTM model can perform well when combined with a CRF layer to improve the NER model results [3], [25], [44].

C. PROPOSED MODEL

For the sake of this research, a deep neural network model based on transfer learning named Pooled-GRU Model with Multilingual Universal Sentence Encoder (Pooled-GRU) is developed to tackle the problem of Arabic named entities extraction out of text. As depicted in **Figure 3**, the Pooled-GRU model consists of six layers as follows:

1) **Text embedding:** is one of the main steps of any NLP tasks, as they allow words of similar meanings have closely similar representations. Different algorithms were implemented for word-based embedding like GloVe [45] or character-based embedding like FastText [46]. Recently, more advanced research for text semantics representation was introduced. Text is represented based on a transformer attentive architecture for text contextual representations [47] such as: semi-supervised Learning [48], ULMFit [49], ELMo [50], Bidirectional Encoder Representations from Transformers (BERT) model [51], and Universal Sentence Encoder (USE) [11].
In this research, we have used a pre-trained sentence (or sequence) embedding algorithm named Universal Sentence Encoder (USE). USE is a sentence encoding model released by Google in July 2018 that aims to provide sentence-level embedding rather than word or character level embedding (See **Figure 4**). USE model was implemented using two techniques (i) transformer-based contextual representation of sentences, and (ii) deep averaging network (DAN) [52] for sentence to sentence similarity representation. USE was first introduced for English [11], [53] and then was implemented for multi-languages including Arabic (MUSE) [54] which is used in this research.
2) **Bi-Directional-GRU:** in this second layer the Bidirectional-Gated Recurrent Unit (BiGRU) [55] as an enhanced version of a Recurrent Neural Network (RNN) is used. The GRU proposes a solution of gradient vanishing problem using two gates (reset and update gate) with a training time relatively faster than Long Short Term Memory (LSTM) [56] which consists of three gates (input, output, and forget gate). The mathematical equations for the GRU are as follows:

$$s_t = (1 - u_t) \odot s_{t-1} + u_t \odot \tilde{s}_t \tag{1}$$

where

$$\tilde{s}_t = tanh(W_s x_t + r_t \odot (z_s s_{t-1}) + b_s) \tag{2}$$
$$u_t = \sigma(W_u x_t + z_u s_{t-1} + b_s) \tag{3}$$
$$r_t = \sigma(W_r x_t + z_r s_{t-1} + b_s) \tag{4}$$

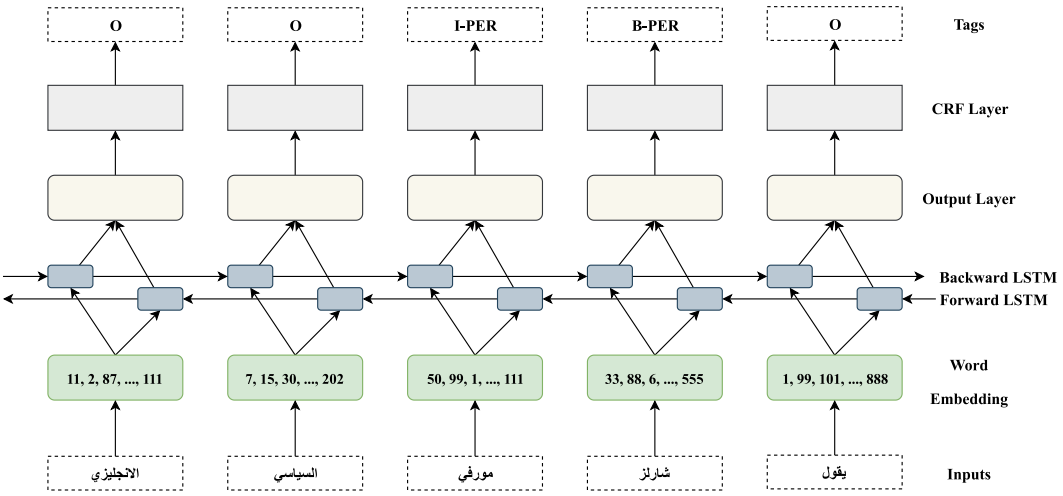

FIGURE 2. Proposed architecture for the Bi-LSTM-CRF model [3].

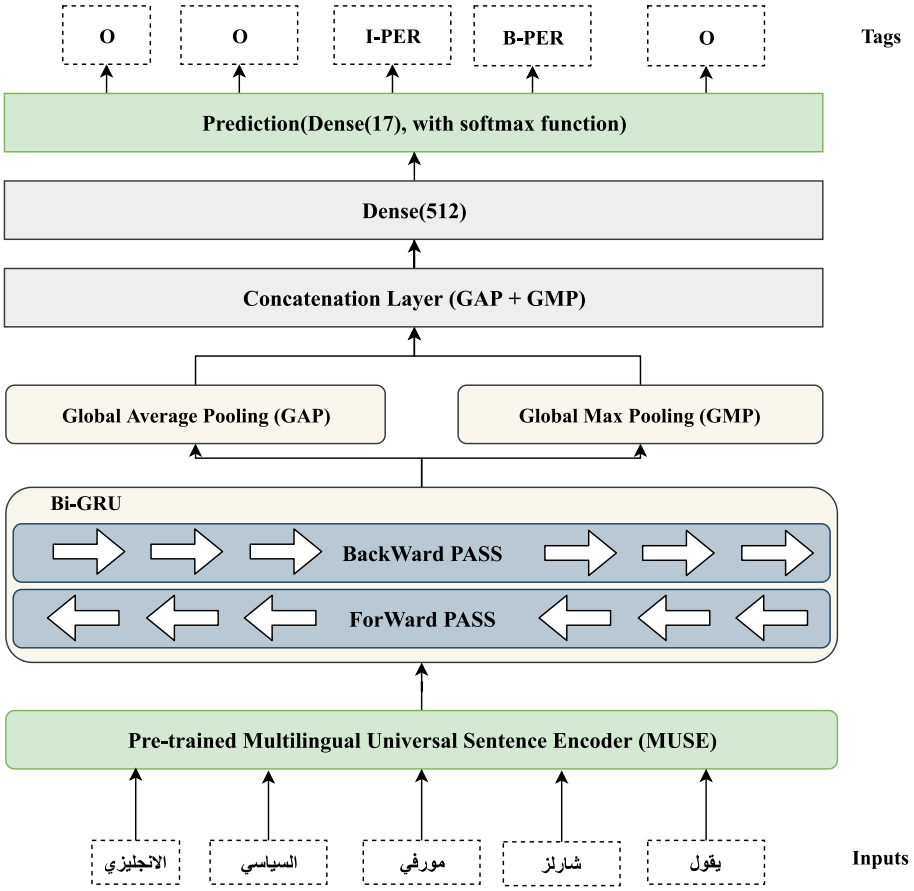

FIGURE 3. Proposed architecture for the Pooled-GRU model.

The update gate is computed using equation u_t (3), the reset gate is computed using equation r_t (4), and the current memory content is represented using equation $\tilde{s}_t$ (2). $W_s, W_u, W_r, Z_s\ Z_u, Z_r, b_s$ represent the Weight matrices, x_t represents the vector input to the time-step t, st equation (1) represents the final current exposed hidden state, and $\odot$ represents the element-wise multiplication.

3) **Pooling layer:** the Global Average Pooling (GAP) and Global Max Pooling (GMP) are used in this layer to extract the discriminative features of the input text and retain them to the next layers of the network.
4) **Concatenation layer:** although its evident in research that GAP helps the model to achieve better results than using GMP since in the GAP loss, all the discriminative features are averaged and retained to the next layer of the network, whereas, in the GMP loss only the maximum values of the features are retained to the next layer [57]. We decided to compute both values of GAP and GMP and concatenate them in a single vector to keep all the possible discriminative features of the text.
5) **Fully connected network**: a Dense layer of 512 fully connected neurons with Rectified Linear Unit (ReLU) are then used to learn the discrimantive features of the text out of the concatenation of the pooling GAP and GMP layer. According to [58], ReLU activation function has less computation cost (i.e. space and time) when compared to other activation functions such as

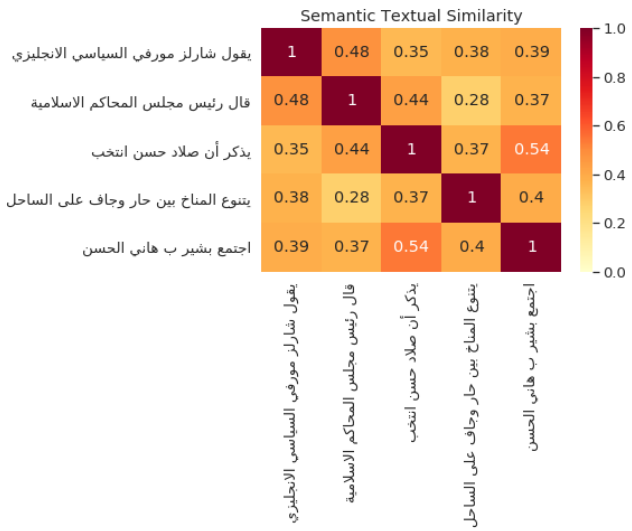

FIGURE 4. Sentence similarity using USE.

sigmoid as the later involve an exponential function which is more computational expensive than the Rectified Linear function. Moreover, ReLU is used to prevent the vanishing gradient problem during the network training.

6) **Classification layer:** a Dense layer of 512 fully connected neurons with softmax activation function for input classification. As the dataset is annotated using the IOB framework and we have 8 coarse-grained classes (B-Class and I-Class) and in addition to the O-Class will lead to 17 classes for classifying an input word.

The Pooled-GRU model is designed based on the transfer learning architectures among deep neural networks [59]–[62]. Transfer learning has been widely used in image classification and computer vision [63]. However, since the advent of transformer based deep neural networks for text representation [47], [48], [51], [54], transfer learning models for text learning and classification has appeared. For instance, the USE model is designed as an independent classifier and can be used in different natural language processing (NLP) applications including our research problem (i.e. NER). The USE was trained on a large multi-lingual corpora to represent the semantic relationships among sentences and phrases and then to utilize this representation along with the transformer architecture [47], [48] to tackle different NLP research problems. The reader is referred to [11], [53], [54] for more details of how USE can be applied in NLP research. For the sake of this research, and based on the transfer learning architecture we transferred the knowledge represented by the USE model to the upper part of the model (Pooled-GRU) to better learn and classify the Arabic named entities in the $\text{WikiFANE}_{\text{Gold}}$ dataset.

IV. EXPERIMENTATION SETUP AND RESULTS

In this section, we discuss the experimentation setup and the parameters that were used to train and evaluate the proposed models discussed earlier in Section III. The following subsections discuss the evaluation measures that were used to evaluate the proposed models and highlight the models' evaluation results.

A. EXPERIMENTATION SETUP

In order to evaluate the proposed deep models (i.e. Baseline: Bi-LSTM-CRF [3] and Pooled-GRU with MUSE). both of them were trained and evaluated using the same train and testing datasets (See Section III-A). As presented in Table **3** the models were trained for 25 epochs with a learning rate of 0.1 and a patch size of 256. The RMSprop[1] optimizer was used for the baseline model training whereas the Adam[2] optimizer was used for the Pooled-GRU with MUSE one. An embedding size of 300 was used to represent the input text to the baseline model whereas an embedding vector of 512 was used to represent the input text for the Pooled-GRU with MUSE one. Finally, both models were implemented using Keras a python-based deep learning framework.[3]

TABLE 3. Proposed models hyper-parameters.

Model	Bi-LSTM-CRF	Pooled-GRU + MUSE
Number of parameters	625k	658,705
Number of epochs	25	25
Batch size	256	256
Optimizer	RMSprop	Adam
Learning rate	0.1	0.1
Embedding vector size	300	512

Deep neural networks are usually trained with overfitting especially when the training dataset is small. Although the training dataset in our research is relatively large with 300K tokens (see Section III-A), we have used the callback function of 'EarlyStopping[4]' from Keras callbacks to stop the model training when the computed validation loss value is stopped improving. The validation loss value is computed in each training epoch.

B. EVALUATION MEASURES

To evaluate the performance of our proposed models we used different measures. We first measured the **Accuracy**, which is a well known measure to evaluate any machine or deep learning model. The accuracy measure can be computed using the following equation 5:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Secondly, the **Precision** measure which represents the number of entities that the model predicted correctly out of the overall predicted entities, was used. The precision measure can be computed using equation 6:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Third, the **Recall** measure, which represents the number of entities that the model predicted correctly out of the overall entities in the dataset. The recall measure can be computed using equation 7:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where, TP refers to True Positives, TN refers to True Negatives, FP refers to False Positives, and FN refers to False Negatives.

Finally, **F1-measure** can be computed based on the precision and recall measures using equation 8:

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{8}$$

[1]https://keras.io/optimizers/#rmsprop
[2]https://keras.io/optimizers/#adam
[3]https://keras.io/
[4]https://keras.io/callbacks/

TABLE 4. The performance results of the proposed Bi-LSTM-CRF model [3].

	Precision	Recall	F1-Score	Accuracy
PER	70.50%	75.05%	72.05%	
ORG	69.34%	77.45%	73.11%	
LOC	71.50%	76.25%	74.35%	
Overall	70.00%	76.05%	73.00%	75.73%

TABLE 5. The performance results of the proposed Pooled-GRU model.

	Precision	Recall	F1-Score	Accuracy
PER	71%	69%	70%	
ORG	60%	59%	60%	
LOC	65%	76%	71%	
GPE	79%	62%	69%	
PRO	59%	47%	53%	
FAC	61%	31%	41%	
VEH	85%	38%	53%	
WEA	60%	23%	33%	
O	94%	98%	96%	
Overall	90%	91%	**90**%	91.20%

C. RESULTS AND FINDINGS

As presented in Table 4 and based on the reported results in our previous work [3], the baseline: Bi-LSTM-CRF model achieved an overall F1= 73.00% for the classification of the tree categories Person (F1 = 72.05%), Organization (F1 = 73.11%), and Location (F1 = 74.35%).

Focusing on our proposed model (i.e. Pooled-GRU model) and as presented in Table 5 the pooled-GRU model outperforms the Bi-LSTM-CRF model with around 17% enhancement based on the F1 measure. The model was also trained on the whole 8 coarse-grained classes instead of three classes in comparison to the baseline model. The model achieved an overall F1 = 90.25% and overall accuracy of 91.20%.

V. DISCUSSION

In order to show the significance of the proposed Pooled-GRU transfer learning model for Arabic NER, the model results are evaluated against other state-of-the-art related work. The Pooled-GRU model results are compared to our previous work Bi-LSTM-CRF [3] results. To the best of our knowledge our previous work and this work are the only deep learning models evaluated using the $\text{WikiFANE}_{\text{Gold}}$ dataset [12] used in this research. As the related work model was evaluated using only three classes (namely person, organization, and location), we will discuss the results using only three of them.

As presented in Table 6, the Pooled-GRU model outperforms the Bi-LSTM-CRF model in all of the reported three classes. More precisely, the Pooled-GRU achieved an F1 = 80% for the Person class in comparison to F1 = 72.05% that was achieved by the Bi-LSTM-CRF model, an F1 = 75% for the Organization class in comparison with F1 = 73.11% that was achieved by the Bi-LSTM-CRF model, and an F1 = 75% for the Location class in comparison with F1 = 74.35% that was achieved by the Bi-LSTM-CRF model. Based on these results, it can be seen that the lowest

TABLE 6. The performance results of the proposed Pooled-GRU model.

Model F1 Results	Bi-LSTM-CRF	Pooled-GRU
PER	72.05%	80.00%
ORG	73.11%	75.00%
LOC	74.35%	75.00%

TABLE 7. The performance results of the MUSE Without Pooled-GRU.

	Precision	Recall	F1-Score	Accuracy
PER	63%	72%	67%	
ORG	68%	35%	46%	
LOC	60%	63%	61%	
GPE	72%	75%	73%	
PRO	58%	37%	45%	
FAC	56%	37%	44%	
VEH	63%	50%	55%	
WEA	55%	23%	32%	
O	89%	91%	90%	
Overall	87%	85%	**86**%	88.71%

enhancement was in the class of Location, whereas the highest enhancement was in the class of Person. This finding can be explained due to the dataset distribution on the classes and their sizes. Going back to Figure 1, we can see that the Person has the highest size among the other two with 20916 named entities whereas the Location class has the lowest among them with 4875 named entities.

According to [64] transfer learning can be more beneficial for datasets with small number of labels. By comparing the results achieved by our transfer learning Pooled-GRU model between the case when it was trained on the three main classes only (see Table 6) to the results achieved on the same classes but when it was trained on the whole coarse-grained ones (see Table 5), it can be seen that the model performance was enhanced with 10% for the Person class, 15% for Organization class, and 4% for the Location class on the F1 results when was trained on small number of classes.

The advancement in the research results can be explained in two points:

- The capabilities of a the MUSE in learning semantic representations among sentences and phrases. To prove this point of view, we have used the USE alone as a classifier for the same dataset and as presented in Table 7, the USE alone was able to achieve an overall F1 = 86%. By focusing on the main three classes, the MUSE achieved an F1 = 67% for the Person class, F1 = 46% for the Organization class, and F1= 61% for the Location class. Although the results of the MUSE for classifying the tree main classes (Person, Organization, and Location) are lower than the achieved results by the transfer learning model (i.e. Pooled-GRU), it can be seen that using MUSE alone to classify some other classes such Geopolitical one is higher than the Pooled-GRU (F1 = 73% for MSUE vs. 69% for the Pooled-GRU). Therefore, more work on enhancing the recall of the Pooled-GRU model should be conducted in upcoming research.

- The significance of the proposed transfer learning model based on the Pooled-GRU in classifying Arabic named entities. By transferring the knowledge from the MUSE network to the Pooled-GRU one, an enhancement of 4% on the overall F-measure was achieved.

VI. CONCLUSION AND FUTURE WORK

In this study, we investigated the impact of the development of a transfer learning model Pooled-GRU model over our previous Bi-LSTM-CRF model for Arabic named entity recognition. Both models were tested on the $\text{WikiFANE}_{\text{Gold}}$ dataset, and the results showed great behavior for our new proposed model with about 17% enhancement over the F-measure. On the other hand, proven accuracy levels where obtained in our newly proposed models 91.20% in comparison to 75.73% in our previous work. Our work is significant in results and performance when compared to the recently developing ANER systems. Research findings go in line with literature [64] as we shown the improved performance of the Pooled-GRU model when transfer learning is used.

REFERENCES

[1] K. Shaalan, "A survey of Arabic named entity recognition and classification," *Comput. Linguistics*, vol. 40, no. 2, pp. 469–510, Jun. 2014.
[2] R. E. Salah and L. Q. B. Zakaria, "A comparative review of machine learning for Arabic named entity recognition," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 7, no. 2, p. 511, Apr. 2017.
[3] S. D. A. Alzboun, S. K. Tawalbeh, M. Al-Smadi, and Y. Jararweh, "Using bidirectional long short-term memory and conditional random fields for labeling Arabic named entities: A comparative study," in *Proc. 5th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS)*, Oct. 2018, pp. 135–140.
[4] D. Kaur and V. Gupta, "A survey of named entity recognition in English and other Indian languages," *Int. J. Comput. Sci. Issues*, vol. 7, no. 6, p. 239, 2010.
[5] R. Grishman, "The NYU system for MUC-6 or where's the syntax?" in *Proc. 6th Message Understand. Conf. (MUC-6)*, Columbia, MD, USA, Nov. 1995.
[6] T. Wakao, R. Gaizauskas, and Y. Wilks, "Evaluation of an algorithm for the recognition and classification of proper names," in *Proc. 16th Conf. Comput. Linguistics*, vol. 1, 1996, pp. 418–423.
[7] G. Petasis, F. Vichot, F. Wolinski, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Using machine learning to maintain rule-based named-entity recognition and classification systems," in *Proc. 39th Annu. Meeting Assoc. Comput. Linguistics*, 2001, pp. 426–433.
[8] D. N. Shah and H. B. Bhadka, "Named entity recognition from Gujarati text using rule-based approach," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2017, pp. 797–805.
[9] R. Alfred, L. C. Leong, C. K. On, and P. Anthony, "Malay named entity recognition based on rule-based approach," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 3, pp. 300–306, 2014.
[10] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," 2018, *arXiv:1812.09449*. [Online]. Available: http://arxiv.org/abs/1812.09449
[11] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," 2018, *arXiv:1803.11175*. [Online]. Available: http://arxiv.org/abs/1803.11175
[12] F. Alotaibi and M. Lee, "A hybrid approach to features representation for fine-grained Arabic named entity recognition," in *Proc. 25th Int. Conf. Comput. Linguistics (COLING)*, 2014, pp. 984–995.
[13] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 473–480.
[14] S. Morwal, "Named entity recognition using hidden Markov model (HMM)," *Int. J. Natural Lang. Comput.*, vol. 1, no. 4, pp. 15–23, Dec. 2012.
[15] S. Morwal and N. Jahan, "Named entity recognition using hidden Markov model (HMM): An experimental result on Hindi, Urdu and Marathi languages," *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, vol. 3, no. 4, pp. 1–5, 2013.
[16] H. L. Chieu and H. T. Ng, "Named entity recognition: A maximum entropy approach using global information," in *Proc. 19th Int. Conf. Comput. Linguistics*, vol. 1, 2002, pp. 1–7.
[17] H. L. Chieu and H. T. Ng, "Named entity recognition with a maximum entropy approach," in *Proc. 7th Conf. Natural Lang. Learn. HLT-NAACL*, 2003, pp. 160–163.
[18] G. Paliouras, V. Karkaletsis, G. Petasis, and C. D. Spyropoulos, "Learning decision trees for named-entity recognition and classification," in *Proc. ECAI Workshop Mach. Learn. Inf. Extraction*, 2000, pp. 1–6.
[19] G. Szarvas, R. Farkas, and A. Kocsor, "A Multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms," in *Proc. Int. Conf. Discovery Sci.* Berlin, Germany: Springer, 2006, pp. 267–278.
[20] A. Ekbal and S. Bandyopadhyay, "Named entity recognition using support vector machine: A language independent approach," *Int. J. Elect., Comput., Syst. Eng.*, vol. 4, no. 2, pp. 155–170, 2010.
[21] Y.-C. Wu, T.-K. Fan, Y.-S. Lee, and S.-J. Yen, "Extracting named entities using support vector machines," in *Proc. Int. Workshop Knowl. Discovery Life Sci. Literature*. Berlin, Germany: Springer, 2006, pp. 91–103.
[22] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," 2016, *arXiv:1603.01354*. [Online]. Available: http://arxiv.org/abs/1603.01354
[23] P.-H. Li, R.-P. Dong, Y.-S. Wang, J.-C. Chou, and W.-Y. Ma, "Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2664–2669.
[24] J. Yang, Y. Zhang, and F. Dong, "Neural reranking for named entity recognition," 2017, *arXiv:1707.05127*. [Online]. Available: http://arxiv.org/abs/1707.05127
[25] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, *arXiv:1603.01360*. [Online]. Available: http://arxiv.org/abs/1603.01360
[26] Q. Tran, A. MacKinlay, and A. J. Yepes, "Named entity recognition with stack residual LSTM and trainable bias decoding," 2017, *arXiv:1706.07598*. [Online]. Available: http://arxiv.org/abs/1706.07598
[27] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," 2016, *arXiv:1603.06270*. [Online]. Available: http://arxiv.org/abs/1603.06270
[28] F. Souza, R. Nogueira, and R. Lotufo, "Portuguese named entity recognition using BERT-CRF," 2019, *arXiv:1909.10649*. [Online]. Available: http://arxiv.org/abs/1909.10649
[29] W. Karaa and T. Slimani, "A new approach for Arabic named entity recognition," *Int. Arab J. Inf. Technol. (IAJIT)*, vol. 14, no. 3, pp. 1–7, 2017.
[30] F. Dahan, A. Touir, and H. Mathkour, "First order hidden Markov model for automatic Arabic name entity recognition," *Int. J. Control Automat.*, vol. 123, no. 7, pp. 37–40, Aug. 2015.
[31] S. Al-Shoukry and N. Omar, "Proper nouns recognition in Arabic crime text using machine learning approach," *J. Theor. Appl. Inf. Technol.*, vol. 79, no. 3, pp. 506–513, 2015.
[32] Y. Benajiba and P. Rosso, "Arabic named entity recognition using conditional random fields," in *Proc. Workshop HLT NLP Arabic World LREC*, vol. 8, 2008, pp. 143–153.
[33] S. AbdelRahman, M. Elarnaoty, M. Magdy, and A. Fahmy, "Integrated machine learning techniques for Arabic named entity recognition," *Int. J. Comput. Sci. Issues*, vol. 7, pp. 27–36, Jul. 2010.
[34] N. F. Mohammed and N. Omar, "Arabic named entity recognition using artificial neural network," *J. Comput. Sci.*, vol. 8, no. 8, pp. 1285–1293, Aug. 2012.
[35] Y. Benajiba, M. Diab, and P. Rosso, "Arabic named entity recognition: An SVM-based approach," in *Proc. Arab Int. Conf. Inf. Technol. (ACIT)*. Amman, Jordan: Assoc. Arab Univ., 2008, pp. 16–18.
[36] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using word representations," 2018, *arXiv:1804.05630*. [Online]. Available: http://arxiv.org/abs/1804.05630
[37] T. Kanan, R. Kanaan, O. Al-Dabbas, G. Kanaan, A. Al-Dahoud, and E. Fox, "Extracting named entities using named entity recognizer for Arabic news articles," *Int. J. Adv. Stud. Comput., Sci. Eng.*, vol. 5, no. 11, pp. 78–84, 2016.

[38] K. K. Shahina, P. V. Jyothsna, G. Prabha, B. Premjith, and K. P. Soman, "A sequential labelling approach for the named entity recognition in Arabic language using deep learning algorithms," in *Proc. Int. Conf. Data Sci. Commun. (IconDSC)*, Mar. 2019, pp. 1–6.
[39] C. Helwe and S. Elbassuoni, "Arabic named entity recognition via deep co-learning," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 197–215, Jun. 2019.
[40] M. Ali, G. Tan, and A. Hussain, "Bidirectional recurrent neural network approach for Arabic named entity recognition," *Future Internet*, vol. 10, no. 12, p. 123, Dec. 2018.
[41] M. N. A. Ali, G. Tan, and A. Hussain, "Boosting Arabic named-entity recognition with multi-attention layer," *IEEE Access*, vol. 7, pp. 46575–46582, 2019.
[42] F. B. Mesmia, K. Haddar, N. Friburger, and D. Maurel, "CasANER: Arabic named entity recognition tool," in *Intelligent Natural Language Processing: Trends and Applications*. Cham, Switzerland: Springer, 2018, pp. 173–198.
[43] O. Kuru, O. A. Can, and D. Yuret, "CharNER: Character-level named entity recognition," in *Proc. 26th Int. Conf. Comput. Linguistics (COLING)*, 2016, pp. 911–921.
[44] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991
[45] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[46] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," 2016, *arXiv:1612.03651*. [Online]. Available: http://arxiv.org/abs/1612.03651
[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[48] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3079–3087.
[49] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*. [Online]. Available: http://arxiv.org/abs/1801.06146
[50] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL*, 2018, pp. 1–15.
[51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[52] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1681–1691.
[53] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2018, pp. 169–174.
[54] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," 2019, *arXiv:1907.04307*. [Online]. Available: http://arxiv.org/abs/1907.04307
[55] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555
[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[57] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929, doi: 10.1109/cvpr.2016.319.
[58] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[59] L. Y. Pratt, J. Mostow, and C. A. Kamm, "Direct transfer of learned information among neural networks," in *Proc. 9th Nat. Conf. Artif. Intell. (AAAI)*, vol. 2, 1991, pp. 584–589. [Online]. Available: http://dl.acm.org/citation.cfm?id=1865756.1865767
[60] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[61] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
[62] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 270–279.
[63] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.
[64] J. M. Giorgi and G. D. Bader, "Transfer learning for biomedical named entity recognition with neural networks," *Bioinformatics*, vol. 34, no. 23, pp. 4087–4094, Dec. 2018.

MOHAMMAD AL-SMADI received the Ph.D. degree in computer science from the Graz University of Technology, in 2012. He is currently an Associate Professor with the Computer Science Department, Jordan University of Science and Technology. He has coauthored several technical articles in established journals and conferences in fields related to *Social and Semantic Computing*, *Knowledge Engineering*, *Natural Language Processing*, *Technology Enhanced Learning*. He is Co-Chairing many IEEE events, such as OSNT, SNAMS, BDSN, iLearn, and many others.

SAAD AL-ZBOON received the B.S. degree in computer science from the Jordan University of Science and Technology, Jordan, in 2014, where he is currently pursuing the master's degree in computer science. His research interests include NLP, machine learning, and semantic computing.

YASER JARARWEH received the Ph.D. degree in computer engineering from The University of Arizona, in 2010. He is currently on unpaid leave with the Jordan University of Science and Technology and a Professor of computer science with Duquesne University. He has coauthored many technical articles in established journals and conferences in fields related to cloud computing, HPC, SDN, and big data. He is Chairing many IEEE events such as AICCSA, SDS, FMEC, ICICS, SNAMS, BDSN, IoTSMS, and many others. He served as a guest editor for many special issues in different established journals. He is also the Steering Committee Chair of the IBM Cloud Academy Conference. He is an Associate Editor of the *Cluster Computing Journal* (Springer), *Information Processing and Management*, and others.

PATRICK JUOLA received the Ph.D. degree in computer science from the University of Colorado at Boulder, in 1995. He is currently a Professor of computer science with Duquesne University, where he directs the Evaluating Variations in Language Laboratory. He specializes in the analysis of texts to determine their authorship, and is best known for his 2013 analysis of The Cuckoo's Calling, a detective novel he revealed to have been written by J. K. Rowling.