Computers, Materials & Continua
DOI:10.32604/cmc.2021.016054
Article

Tech Science Press

Arabic Named Entity Recognition: A BERT-BGRU Approach

Norah Alsaaran* **and Maha Alrabiah**

Department of Computer Science, Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia
*Corresponding Author: Norah Alsaaran. Email: NSAlsaaran@imamu.edu.sa
Received: 20 December 2020; Accepted: 20 January 2021

Abstract: Named Entity Recognition (NER) is one of the fundamental tasks in Natural Language Processing (NLP), which aims to locate, extract, and classify named entities into a predefined category such as person, organization and location. Most of the earlier research for identifying named entities relied on using handcrafted features and very large knowledge resources, which is time consuming and not adequate for resource-scarce languages such as Arabic. Recently, deep learning achieved state-of-the-art performance on many NLP tasks including NER without requiring hand-crafted features. In addition, transfer learning has also proven its efficiency in several NLP tasks by exploiting pretrained language models that are used to transfer knowledge learned from large-scale datasets to domain-specific tasks. Bidirectional Encoder Representation from Transformer (BERT) is a contextual language model that generates the semantic vectors dynamically according to the context of the words. BERT architecture relay on multi-head attention that allows it to capture global dependencies between words. In this paper, we propose a deep learning-based model by fine-tuning BERT model to recognize and classify Arabic named entities. The pre-trained BERT context embeddings were used as input features to a Bidirectional Gated Recurrent Unit (BGRU) and were fine-tuned using two annotated Arabic Named Entity Recognition (ANER) datasets. Experimental results demonstrate that the proposed model outperformed state-of-the-art ANER models achieving 92.28% and 90.68% F-measure values on the ANERCorp dataset and the merged ANERCorp and AQMAR dataset, respectively.

Keywords: Named entity recognition; Arabic; deep learning; BGRU; BERT

1 Introduction

Textual information represents a wide share of digital content, and it is continuing to grow rapidly every moment, which requires linguistic and deep semantic analysis techniques to achieve a better and faster understanding of this information. NER is one of the techniques used to identify and classify each word in a given text into predefined semantic categories such as person name, location name, organization name, and miscellaneous. NER plays an essential role in several NLP tasks such as information retrieval, question answering, machine translation, and text summarization. However, NER is considered a challenging task and has several difficulties. For

example, the presence of homonyms, synonyms and acronyms is causing text ambiguity, which is considered one of the main challenges that face NER. Additionally, a word may appear as a Named Entity (NE) in one context and as a common noun in another; a word may also refer to more than one entity type. Moreover, names compose the largest part of language and they are constantly evolving in different domains, and as NER is considered a domain-dependent task, this requires updating annotated corpora which is costly and requires linguistic and domain expertise [1]. Furthermore, due to the international domination of the English language in several fields such as communications, science, information technology, business, and other fields, most of the NER research is focused on studying English, which caused the lack of variety of text genres for other languages such as Arabic.

There is a considerable amount of work on NER; different techniques relying on rule-based approaches, machine learning-based approaches and hybrid approaches have been introduced. Although these approaches showed significant and successful results, they often require a lot of human effort in designing rules and/or features. Recently, deep learning became a trending approach attracting significant attention due to its state-of-the-art performance in various NLP tasks including NER. The main strength of deep learning is its reliance on non-linear transformations while learning complicated features from data, in addition to reducing the time and effort required to design NER features by learning useful representations from raw data automatically [2]. Recurrent Neural Network (RNN) is a neural network architecture that can handle sequential data with feedback loops that memorize the previous time steps output and use them to predict the current time step input. RNNs are often used in NLP tasks that require the sequential processing of texts, such as translation applications where understanding the next word is based on the context of the previous text [3,4]. Bidirectional RNNs, such as BGRU, can learn long dependency input which makes it superior in many NLP tasks.

NLP tasks especially for low recourse languages such as Arabic suffer from nonsufficient training data, which causes poor model generalization. Transfer learning helps in solving this issue by transferring knowledge across domains or tasks. It aims to extract the knowledge from one or more source tasks and applies that knowledge to other target tasks. This improves the performance of learning and avoid expensive efforts for data labeling. Fine-tuning is one of the transfer learning strategies where the downstream NLP task model is initialized with the pre-trained language model parameters. Then, all the parameters are fine-tuned using the downstream task's labelled data. This requires less time to train the down-stream task model because a huge part was done by the pre-trained language model [5]. Word2vec [6], Glove [7], and Fasttext [8] are all examples of pre-trained language models, called word embeddings, that can be used in transfer learning. Although these word embeddings have achieved great success in many NLP tasks, they are considered as non-contextual language models as they only capture static meanings of words and represent each word by a single vector, which makes polysemous words a challenge [9]. On the other hand, contextual language models have the ability to understand the full context of each word in an input sequence and build more than one vector for each word depending on its position in text. These models have shown that such representations are able to achieve state-of-the-art performance in many complex NLP tasks. BERT is an example of a contextual language model that can be fine-tuned. BERT is based on a multilayer bidirectional transformer encoder that takes into account both left and right contexts [10].

Although the last decade has witnessed a significant progress in ANER, the task of developing a robust ANER system remains challenging compared to other languages. This is due to the fact that Arabic has no capitalization of proper nouns, which is considered a significant feature

in some languages that helps in defining nouns. In addition, Arabic has a complex morphological system that allows conjunctions, prepositions and pronouns to be added to words as suffixes and prefixes, such as the word waletasa'ado "ولتسعدوا" which means "and so that you can become happy." Common Arabic writings often miss the presence of short vowels (diacritics) that are essential for identifying the correct word's form and meaning. For example, the word "كتب" may have several variations which are kataba "كَتَبَ" meaning "wrote", kotob "كُتُب" meaning "books" and "كُتِب" meaning "was written." Moreover, there are variations in writing certain Arabic letters especially in transliterated words, such as the word gram, which can be written as "غرام" or "جرام". Additionally, similarity or inherent disagreement in some Arabic characters may cause orthographical confusion. For example, the variation of the hamza which is commonly written as "ا" instead of "إ", "أ" and "آ". Furthermore, the lack of computational linguistic resources for ANER, as mentioned earlier, poses another challenge to ANER [11].

In this work, we fine-tune the pre-trained BERT language model for Arabic NER using a BGRU-based deep learning model. Pre-trained BERT language model parameters were trained extensively using large unannotated data, to encode information about the language. Vectors representations by the pre-trained BERT language model are fine-tuned on the ANER task-specific dataset rather than using them directly. A BGRU layer was added on top of the pre-trained BERT language model for further training to improve the final context vectors. Fine-tuning helps in training the model with less time and epochs because a huge part of the training was performed by the pre-trained BERT language model. In addition, the model can be trained to good performance with small training dataset because the ANER model weights are initialized with the pre-trained BERT weights.

The rest of this paper is organized as follows. Section 2 provides an overview of the underlying research field. Section 3 presents the proposed BERT-BGRU architecture for ANER in details. Section 4 describes the used datasets, the experimental settings, baselines, and results. Section 5 provides a discussion of the obtained results. Finally, Section 6 presents the conclusions and future work.

2 Related Work

There is a fair amount of literature concerning NER research, which covers a wide range of approaches such as rule-based, machine learning-based, deep learning-based and hybrid approaches. However, deep learning approach has achieved state-of-the-art results using various models, and the Bidirectional Long Short-Term Memory with Conditional Random Fields (BLSTM-CRF) was the most dominating model [12–17]. Different variations of the model were introduced to improve results such as adding a part-of-speech attention mechanism to utilize the part of speech information [12], adding a multi-head attention layer to get the meaning of the words and their syntactic features [13] and using additional word features such as capital letters [14]. Other deep learning models were also applied including using Convolutional Neural Network (CNN). Wang et al. [18] used a hierarchical CNN to extract context information with a gating mechanism into the convolutional layer. Bidirectional Gated CNN were used in [19] for Chinese NER. Kosasih et al. [20] used a BGRU-CRF model for Indonesian NER.

It can be noticed that the success of an NER model depends heavily on its input representation. Word and character embeddings are commonly used; where word-level representations encode the syntactic and semantic features of words and character-level representation helps in dealing with the Out Of Vocabulary (OOV) problem. Zirikly et al. [21] demonstrated that the embedding representation scheme can replace the use of dictionaries and gazetteers and still result

in high performance even if the used training dataset is small. For word embedding, several models were used for the input representation including word2vec [22–24], Glove [25–27], Fastext [28–30], ELMo [31–33] and BERT [34–36]. Whereas, for character embeddings, BLSTM and CNN were the most commonly used models [37–39]. On the other hand, BERT obtained efficient embedding compared to the other models in terms of word and character representation [34,40]. Different layers where added on top of BERT to fine-tune it for NER task; a comparative study demonstrated that fine-tuning BERT with a BLSTM-CRF layer outperformed a linear CRF layer [41]. Another attempt proved that using a BGRU-CRF layer yielded better results [42]. Yan et al. [43] added a multi-head attention layer to the BGRU-CRF layer. Straka et al. [44] combined the representation of FastText, BERT and Flair for Czech NER, which outperformed the BERT only version of their model. Li et al. [41] applied dictionary features and radical features to the fine-tuned BERT model to improve the model performance for Chinese clinical NER.

Regarding ANER, BLSTM-CRF was also a dominating model in the literature; Gridach [45] used a BLSTM-CRF model with a BLSTM-based character-level embedding and a word2vec pre-trained word embedding to extract named entities from Arabic social media. On the other hand, Khalifa et al. [46] was the first ANER model that utilized CNN for constructing character embeddings with BLSTM-CRF and word2vec word embedding. El Bazi et al. [47] used the BLSTM-CRF model with pre-trained Fasttext word embeddings and character-based representations constructed by CNN model. Gridach et al. [48] used character-level representation constructed by BGRU model and pre-trained word2vec word representations to improve their system performance. CRF was used as the tag decoder layer instead of decoding each tag independently. To overcome the problem of lacking sufficient training data, Helwe et al. [49] used deep co-learning which is a semi-supervised learning approach that can be trained using both labeled and unlabeled data. Their model used two classifiers that learn from each other using two different views of the data. Ali et al. [50] added an attention embedding layer that combined the representation of a pre-trained word2vec model and character embeddings provided by a CNN model in order to create a good word representation. BLSTM and BGRU have been compared where BLSTM outperformed BGRU. The authors improved the model by adding a self-attention layer on the top of the encoder in order to provide high or low consideration to words based on their involvement in the creation of the sentence meaning [51]. Ali et al. [52] used BLSTM as encoder and decoder model for ANER; their model outperformed the BGRU-CRF model by [48] and the BLSTM-CRF model by [53].

Fine-tuning transfer learning has been recently investigated in ANER [54], utilized transfer learning with deep neural networks to build a Pooled-GRU model combined with the Multilingual Universal Sentence Encoder (MUSE) language model. Their model outperformed the BLSTM-CRF model proposed by [55]. AraBERT [56] is a BERT model trained specifically for Arabic and it was tested on ANER task. The model outperformed the BLSTM-CRF model proposed by [55] and BERT multilingual.

3 Proposed Model Architecture

Our proposed model consists of three main layers; the first layer produces the input representations using a BERT pre-trained language model. The second layer learns context representation from the previous component output using BGRU as a context encoder model. Finally, the tags prediction layer, takes the context representation as input and produces a sequence of tags corresponding to the input sequence. Fig. 1 shows the model architecture.

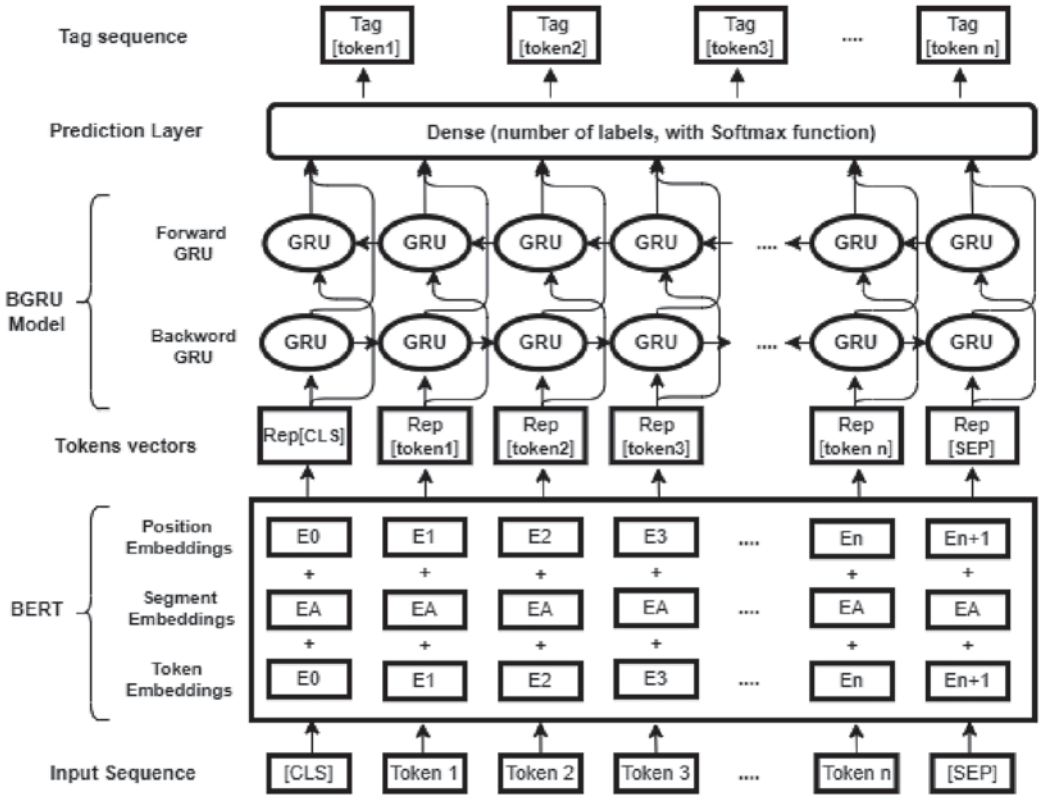

Figure 1: BERT-BGRU model architecture

3.1 Input Representation Layer: BERT Contextual Language Model

BERT is a contextual language model developed by Devlin et al. [10] in 2019, which is based on a multilayer bidirectional transformer encoder that takes into account both left and right contexts. BERT trained using two unsupervised tasks: the masked language model and the next sentence prediction. In the masked language model, randomly 15% of the tokens in the input sequence are masked and the model could predict the target token in a multi-layered context. Then, the final hidden vectors corresponding to the masked tokens are fed into an output Softmax over the vocabulary. The next sentence prediction aims to understand the relationship between two sentences. For every input sequence, the first token is a special classification token [CLS] and the final hidden state corresponding to this token is used as the sequence representation. BERT uses the WordPiece tokenizer by Wu et al. [57], which breaks down any word that is not found in the vocabulary into sub-words to overcome the OOV problem. The final embeddings of BERT are the sum of tokens embedding, segment embedding, and position embedding. Token embedding indicates the embedding of the current word, Segment Embedding indicates the index embedding of the sentence in which the current word is located, Position Embedding indicates the index embedding of the current word position. Two outputs are produces by BERT model; [CLS] sentence embedding vector and word-level embeddings that take context into account.

Since BERT is more suitable for languages that have very rich morphological systems such as Arabic, we exploit AraBERTv0.1 [56] pretrained model as the input representation to the context encoder layer in our proposed ANER system. AraBERTv0.1 is trained using the BERTBASE size,

which has 12 heads, 12 layers, 768 hidden units per layer, and a total of 110M parameters. The corpora used for training the model are the manually scraped Arabic news websites for articles and two publicly available large Arabic corpora: the 1.5 billion words Arabic Corpus and the Open Source International Arabic News Corpus (OSIAN). The final dataset size is 70 million sentences, corresponding to 24 GB of text. Their dataset covers news from different media in different Arab regions, and therefore can be representative of a wide range of topics discussed in the Arab world. In addition, words that include Latin characters were preserved, since it is common to mention named entities, scientific and technical terms in their original language, to avoid information loss. The final size of vocabulary was 64k tokens.

3.2 Context Encoder Layer: BGRU Model

The encoder is used to capture the context and order of words in the sentence and encode them into a fixed length representation. RNN is commonly used as context encoder for NER tasks; it processes the timesteps of its input sequences in order, where shuffling or reversing the timesteps can completely change the representations that the RNN extracts from the sequence. RNN becomes less accurate with long sequence data in which it is difficult for the network to memorize far away from previous time steps outputs [3]. This problem is called the vanishing gradient problem. GRU is a variation of RNN that helps in handling the gradient vanishing problems and can learn long dependency input. It uses update and reset gates to control and update the cell state. The update gate controls what information from the previous time step output should pass and used as the next time step input. The reset gate controls what information from the previous time step output should be forgotten and what information is relevant to the current time step input [3,4]. At time step t, the output of both reset gate r_t and forget gate u_t are calculated using their own weights matrix U_r, U_u, W_r, and W_u. Then, a sigmoid activation function σ is applied which works as a binary-like mask in which information with near zero values is blocked and information with values near one are allowed to pass. The output of the reset gate is used to calculate the memory content h'_t by performing a Hadamard product $\odot$ which is an element-wise multiplication of r_t with the value of the previous hidden state h_{t-1}. Finally, the output of the update gate u_t is used to determine which information from the current memory content will be used as the new hidden state h_t that will be passed to the next unit [58]. The following present the equations used to calculate the hidden state h_t at time step t.

$$u_t = \sigma(U_u \cdot h_{t-1} + W_u \cdot x_t + b_u) \tag{1}$$

$$r_t = \sigma(U_r * h_{t-1} + W_r * x_t + b_r) \tag{2}$$

$$h'_t = tanh(U * (h_{t-1} \odot r_t) + W * x_t + b) \tag{3}$$

$$h_t = (1 - u_t) * h_{t-1} + u_t \cdot h'_t \tag{4}$$

GRU has a more simplified architecture than LSTM, which is also a variation of the standard RNN. The main difference between GRU and LSTM is that GRU has less parameters than LSTM, which makes it faster to train. In addition, LSTM shows better results with large datasets, while GRU has better results with small datasets [3]. The final hidden state of a unidirectional GRU reflects more information about the end of the input sequence than its beginning. In the NER task, it is important to capture prior and posterior information; considering the two sentences: "Ford was born in Detroit" and "Ford Motors company". The word "Ford" in the first sentence should be tagged as a person named entity while in the second sentence it should be tagged as an organization named entity. Note that the word "Ford" appears in the beginning

of both sentences and the encoder will make the prediction before detecting any words in the sentence, which will cause incorrect predictions. BGRU, the bidirectional variant of GRU, solves this problem by parsing the input sequence from both directions. In fact, it has outperformed regular GRU in many NLP tasks [59]. In BGRU, the forward GRU reads the entire source sequence one word at a time from left-to-right and the backward GRU reads the words from right-to-left. At time step t, the output of the forward GRU h_t^f covers the input sequence in the forward direction and the backward GRU h_t^b covers the input sequence coming in the backward direction. Then, the final hidden state h_t will be a concatenation of h_t^f and h_t^b as $h_t = h_t^f \oplus h_t^b$ [58].

3.3 Tag Prediction Layer: Dense Layer with Softmax Function

The prediction layer produces a sequence of tags corresponding to the input sequence. At each time step t, the hidden state h_t fed to a fully connected layer (dense layer) in order to get the score of each tag. The nth entry of this vector represents the score for tag k to word n. Matrix (P) with size $k * n$ will be formed where n is the number of words in the input sequence and k is the number of possible named entity tags each word can have. Then the Softmax function turns these scores into a probability distribution. The probability of assigning a tag y to word x is calculated by the following equation, where Y denotes the set of all possible tags.

$$p\,(y|x) = \frac{e^{score(x,y)}}{\sum_{i=1}^{Y} e^{score(x,yi)}} \tag{5}$$

The tag with highest probability at each word position is chosen such as $y = argmaxP\,(y|x)$. Softmax function considers the NER task as a multi-class classification problem in which the tag for each word is independently predicted based on the context vector without considering its neighbours [2].

4 Experiments

Two experiments are performed to evaluate our proposed BERT-BGRU model; one with ANERCorp dataset only and the other on the merged ANERCorp and AQMAR dataset.

4.1 Dataset

We use two annotated datasets to train and evaluate our model, which are ANERCorp and AQMAR. ANERCorp by Benajiba et al. [60] is an Arabic named entities manually annotated *corpus* that is freely available for research purposes. It contains 150,286 tokens and 32,114 named entities from 316 articles that were selected from different types and different newspapers in order to obtain a *corpus* as generalized as possible. The *corpus* was annotated following the annotation scheme defined in the MUC-6 with IOB tagging. Words tagged with nine classes were used for the annotation: B-PERS, I-PERS, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC, and O. Where PERS denotes a person named entity, LOC denotes a location named entity, ORG denotes an organization named entity, MISC denotes miscellaneous, which is a named entity that does not belong to any of the other classes, and O denotes other words that are not named entities. CONLL training file format [61] is used in which the file is organized in 2 columns; the first column for the words and the second one for the tags. Named entities were distributed as 39% for person, 30.4% for location, 20.6% for organization, and 10% for miscellaneous. On the other hand, AQMAR by Mohit et al. [62] is a *corpus* of Arabic Wikipedia articles annotated with named entity information. It contains about 3,000 sentences from 31 Arabic Wikipedia articles

covering several genres including history, technology, science, and sports. ACE guidelines were followed in identifying the named entities boundaries and tagging. IOB tagging scheme was used to annotate 74,000-tokens with person, location, organization, miscellaneous and other tags. Tab. 1 presents the training, validation, and testing statistics for the ANERCorp dataset. Tab. 2 presents the training, validation, and testing statistics for the merged ANERCorp and AQMAR dataset. Both datasets were splitted ∼80% as training dataset, ∼10% as validation dataset and ∼10% as testing dataset.

Table 1: Training, validation, and testing statistics for ANERCorp dataset

	PER	LOC	ORG	MISC	O	Total
Train	5144	4121	2751	1319	96263	109598
Test	725	487	391	164	12361	14128
Validation	568	423	266	172	10559	11988
Total	6437	5031	3408	1655	119183	

Table 2: Training, validation, and testing statistics for merged ANERCorp and AQMAR dataset

	PER	LOC	ORG	MISC	O	Total
Train	6318	5031	2977	3985	125025	143336
Test	1828	1465	917	1178	36099	41487
Validation	668	437	328	326	13463	15222
Total	8814	6933	4222	5489	174587	

4.2 Data Preprocessing

Since ANERCorp and AQMAR do not follow the same tagging scheme. AQMAR dataset uses MIS and PER tags to denote person and miscellanies named entities, respectively. While ANERCorp uses MISC and PERS to denote them. Therefore, we performed the necessary preprocessing to unify the tags before merging them. In addition, and in order to achieve high accuracy results, the dataset was cleaned by removing punctuations, other special characters, and diacritics signs (ً,ٌ,ٍ,َ,ُ,ِ,ّ,ْ). To use the pre-trained BERT model, input sentences should be preprocessed in the same way BERT model was trained. Therefore, we enclosed each input sentence between [CLS] and [SEP] tokens. In addition, words in the input sentence are tokenized into list of tokens that are available in the BERT vocabulary. Out of vocabulary words are progressively split into sub-tokens. We assigned the word's tag for the first sub-token only and we considered the rest of the sub-tokens as padding. For example, the word 'الشامخة' is tokenized as ['الشامخ', '##ة'] and its given tags would be ['O', 'PAD']

4.3 Baseline

To evaluate our BERT-BGRU model, we compare it to five state-of-the-art deep learning-based ANER models that were recently published. The first baseline is the BRGU-Softmax model by Ali et al. [50], which used a pre-trained word2vec embedding layer, an attention embedding layer and a CNN-based character embedding layer. The second baseline is the fine-tuned pre-trained AraBERT model for ANER by Antoun et al. [56]. The third baseline is the BLSTM-CRF

model proposed by [46] using word2vec word embedding and a CNN-based character embedding layer. The fourth baseline [51] is an improvement of [50] that used a BLSTM-Softmax model and a Self-Attention layer on the top of the encoder. Finally, the fifth baseline is a variation of [50] with a BLSTM encoder–decoder model [52]. The first three baselines are trained and evaluated on ANERCorp and the rest are trained and evaluated on the merged ANERcorp and AQMAR dataset.

4.4 Experiment Setting

In this experiment, Pytorch API was used for model implementation and the whole experiment was run on the Google Colab platform (https://colab.research.google.com/) with a Tesla T4 GPU. We utilized the training dataset for training our model and exploited the validation dataset to choose the hyper-parameters. A maximum input sequence length is set to 100, sequences greater than this length would be truncated and sequences less than this length would be padded to obtain the same length. The GRU hidden units is set to 384 which is half of the BERT hidden units. The batch size is 32 for training and 8 for testing and validation. The Adam optimizer was used with 0.1 Warmup and 1e−4 learning rate. The Model is trained for 5 epochs with 0.2 dropout rate to avoid overfitting.

4.5 Results

Due to the random initialization for weights in neural networks, training neural networks is considered a non-deterministic process. For that, we trained our model for 5 times and used the average evaluation indicators; precision, recall, and F-measure for robust evaluation. Precision, Eq. (6), measures the correctness and accuracy of the entities identified by the NER system. While recall, Eq. (7), measures the ability of the system to detect all named entities in a given corpus. Since there is a trade-off between precision and recall, the F-measure, Eq. (8), is used to balance the antagonistic relation between them using the parameter β. Where precision and recall are given equal weights when β is equal to one, precision is favoured when β is greater than one, and recall is favoured when β is smaller than one [2,58].

$$\text{Precision} = \frac{Number\ of\ correct\ named\ entities\ recognized\ by\ the\ model}{Total\ number\ of\ named\ entities\ recognized\ by\ the\ model} \tag{6}$$

$$\text{Recall} = \frac{Number\ of\ correct\ named\ entities\ recognized\ by\ the\ model}{Total\ number\ of\ actual\ named\ entities\ in\ the\ corpus} \tag{7}$$

$$\text{F-measure} = \frac{(\beta^2+1)(Precision * Recall)}{\beta^2(Precision + Recall)} \tag{8}$$

Table 3: Results of BERT-BGRU model on ANERCorp dataset

	PER (%)	LOC (%)	ORG (%)	MISC (%)	O (%)	Overall (%)
P	97.87	93.76	88.03	81.95	99.38	92.20
R	97.07	95.60	90.16	79.82	99.32	92.40
F	97.47	94.67	89.08	80.80	99.36	92.28

Two experiments are performed, one run on ANERCorp only and the other run on the merged ANERCorp and AQMAR dataset. The details of the evaluation results of the five entities identified by the BERT-BGRU model are shown in Tabs. 3 and 4. 'P', 'R', and 'F' denote precision, recall, and macro-averaged F1-score, respectively.

Table 4: Results of BERT-BGRU model on the merged ANERCorp and AQMAR dataset

	PER (%)	LOC (%)	ORG (%)	MISC (%)	O (%)	Overall (%)
P	96.56	92.27	87.95	84.06	98.64	91.90
R	94.68	90.73	84.81	78.37	99.14	89.55
F	95.61	91.49	86.34	81.10	98.89	90.69

5 Discussion

To evaluate our results, we compare the performance of our BERT-BGRU model against the state-of-the-art baselines. To compare our work with [50,56], we assumed that the authors identified the conventional named entities categories used in the literature, which are person, location, organization and miscellaneous since this information is not specified in their paper. To compare our work to [46], we identified person, location, and organization named entities, and to compare our work to [51,52] we identified person, location, organization, and other named entities as stated by the authors. In addition, we used the dataset splits provided by the authors when available and the standard 80% as training dataset, 10% as validation dataset and 10% as testing dataset otherwise. Moreover, to compare our model with [52], we reran the model on the merged ANERCorp and AQMAR datasets that is split into 70% as training dataset, 10% as validation dataset and 20% as testing dataset. The performance of our model against the performance of the first [50] and the second [56] baselines is illustrated in Tab. 5. The performance of our model compared to baseline [46] on the ANERCorp dataset is shown on Tab. 6. On the other hand, Tabs. 7 and 8 illustrate the performance of our model against the fourth [51] and the fifth [52] baselines, respectively, on the merged ANERCorp and AQMAR dataset.

Table 5: Performance of our BERT-BGRU model and baselines model on ANERCorp dataset

	P (%)	R (%)	F (%)
Our model	90.40	90.66	**90.51**
Ali et al. [50]	87.73	86.51	87.12
Antoun et al. [56]	–	–	84.20

**Note*: The values in bold indicate the highest result.

It can be noticed from Tab. 5 that our BERT-BGRU model outperformed [50] by 3.39 F-measure points, which indicates that the context semantic representation of dynamically generated word vectors by the pre-trained BERT language model is better than the non-contextual word vectors representations in representing sentence features. Our model also outperformed [56] by 6.31 points, which reflects the benefit of transferring the knowledge represented by BERT to BGRU deep learning model to help in learning more context information. Furthermore, as shown in Tab. 6, our model outperformed Khalifa et al. [46] by 6.78 F-measure points even though

that they used a pretrained word2vec model with a vocabulary size of 6,261,756 words, while the pretrained BERT model used in our work have a vocabulary size of only 64k words. On the other hand, our model outperformed [51] by 1.77 F-measure points, Tab. 7, and showed comparable results to [52], Tab. 8, with an increase of 0.06 F-measure points. However, [51,52] applied an embedding attention layers on top of the word and character level representations in order to determine how to consolidate the information for each word, however, no such layers were applied in our model since this was already achieved by the transformer multi-head attention in the BERT model architecture. In addition, our model required 5 epochs to train while the model in [52] was trained for 20 epochs. This demonstrates the effectiveness of fine-tuning pre-trained BERT in reducing the required training time for the downstream task model compared to the needed time to train the model from scratch.

Table 6: Performance of our BERT-BGRU model and baselines model on ANERCorp dataset across Person, Location and Organization named entities

	P (%)	R (%)	F (%)
Our model	93.22	94.28	**93.74**
Khalifa et al. [46]	–	–	86.96

Note: The values in bold indicate the highest result.

Table 7: Performance of our BERT-BGRU model and Ali et al. [51] model on merged ANER-Corp and AQMAR dataset

	P (%)	R (%)	F (%)
Our model	93.86	92.34	**93.08**
Ali et al. [51]	91.17	91.54	91.31

Note: The values in bold indicate the highest result.

Table 8: Performance of our BERT-BGRU model and Ali et al. [52] model on merged ANER-Corp and AQMAR dataset

	P (%)	R (%)	F (%)
Our model	92.83	91.35	**92.07**
Ali et al. [52]	93.52	90.54	92.01

Note: The values in bold indicate the highest result.

6 Conclusion

In this paper, we investigate the use of contextualized embeddings methods, namely BERT. By utilizing these embeddings as input to a BGRU deep neural network. Our experiments show the effectiveness of fine-tuning pre-trained BERT language model for languages with rich morphology and low resources specifically in NER task. Furthermore, fine-tuning BERT as the underlying input representation model effectively reduces the training time of the downstream deep neural network-based model since a major part of training was already done by BERT. Moreover, BERT stacks multiple layers of attention making it able to produce rich representations without the need of adding extra embeddings attention layers to the NER model. Our proposed BERT-BGRU

model outperformed the compared baselines and achieved state-of-the-art results on ANERCorp dataset and the merged ANERCorp and AQMAR dataset by achieving an F-measure values of 92.28% and 90.68%, respectively without the need of any feature engineering. Our future work will be committed to using BERT-BGRU-CRF for ANER task in addition to applying additional features such as dictionary features to improve our result.

Acknowledgement: The authors extends their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding and supporting this work through Graduate Students Research Support Program.

Funding Statement: This research is funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University through the Graduate Students Research Support Program.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Zitouni, *Natural Language Processing of Semitic Languages*, Berlin, Heidelberg: Springer, 2014.

[2] J. Li, A. Sun, J. Han and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, early access, 2020.

[3] P. Goyal, S. Pandey and K. Jain, *Deep Learning for Natural Language Processing*, Berkeley, CA: Apress, 2018.

[4] F. Chollet, *Deep Learning with Python*, USA: Manning Publications Co., 2017.

[5] S. Ruder, M. Peters, S. Swayamdipta and T. Wolf, "Transfer learning in natural language processing," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Tutorials*, Minneapolis, Minnesota, pp. 15–18, 2019.

[6] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, Arizona, USA, pp. 1301–3781, 2013.

[7] J. Pennington, R. Socher and C. Manning, "GloVe: Global vectors for word representation," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1532–1543, 2014.

[8] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, no. 1, pp. 135–146, 2017.

[9] Y. Wang, Y. Hou, W. Che and T. Liu, "From static to dynamic word representations: A survey," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1611–1630, 2020.

[10] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, PA, USA, pp. 4171–4186, 2019.

[11] K. Shaalan, "A survey of Arabic named entity recognition and classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469–510, 2014.

[12] S. Zhang, Y. Sheng, J. Gao, J. Chen, J. Huang *et al.,* "A multi-domain named entity recognition method based on part-of-speech attention mechanism," in *Computer Supported Cooperative Work and Social Computing*, Singapore: Springer, pp. 631–644, 2019.

[13] X. Liu, N. Yang, Y. Jiang, L. Gu and X. Shi, "A parallel computing-based deep attention model for named entity recognition," *Journal of Supercomputing*, vol. 76, no. 2, pp. 814–830, 2020.

[14] C. Ronran and S. Lee, "Effect of character and word features in bidirectional LSTM-CRF for NER," in *Proc. of the IEEE Int. Conf. on Big Data and Smart Computing*, Busan, Korea (South), pp. 613–616, 2020.

[15] P. Bhatia, K. Arumae and B. Celikkaya, "Towards fast and unified transfer learning architectures for sequence labeling," in *Proc. ICMLA*, Boca Raton, FL, USA, pp. 1852–1859, 2019.
[16] G. Dekhili, N. Le and F. Sadat, "Improving named entity recognition with commonsense knowledge pre-training," *Knowledge Management and Acquisition for Intelligent Systems*, vol. 11669, pp. 10–20, 2019.
[17] G. Jin and Z. Yu, "A Korean named entity recognition method using Bi-LSTM-CRF and masked self-attention," *Computer Speech & Language*, vol. 65, pp. 101134, 2021.
[18] C. Wang, W. Chen and B. Xu, "Named entity recognition with gated convolutional neural networks," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Cham: Springer, pp. 110–121, 2017.
[19] T. Zhao, H. Liu, Q. Wu, C. Sun, D. Zhan *et al.,* "BiGCNN: Bidirectional gated convolutional neural network for Chinese named entity recognition," in *Database Systems for Advanced Applications*, Cham: Springer, pp. 502–518, 2010.
[20] J. Kosasih and M. Khodra, "Transfer learning for Indonesian named entity recognition," in *Proc. SAIN*, Yogyakarta, Indonesia, pp. 173–178, 2018.
[21] Zirikly and Diab, "Named Entity Recognition for Arabic Social Media," in *Proc. of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, pp. 176–185, 2015.
[22] P. Zhou, S. Zheng, J. Xu, Z. Qi and H. Bao, "Joint extraction of multiple relations and entities by using a hybrid neural network," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Cham: Springer, pp. 135–146, 2017.
[23] D. Liu and X. Zou, "Sequence labeling of Chinese text based on bidirectional GRU-CNN-CRF model," in *Proc. ICCWAMTIP*, Chengdu, China, pp. 31–34, 2018.
[24] C. Huang, Y. Chen and Q. Liang, "Attention-based bidirectional long short-term memory networks for Chinese named entity recognition," in *Proc. of the 2019 4th Int. Conf. on Machine Learning Technologies*, New York, NY, USA, pp. 53–57, 2019.
[25] H. Fei, Y. Ren and D. Ji, "Dispatched attention with multi-task learning for nested mention recognition," *Information Sciences*, vol. 513, no. 1, pp. 241–251, 2020.
[26] M. Xiaofeng, W. Wei and X. Aiping, "Incorporating token-level dictionary feature into neural model for named entity recognition," *Neurocomputing*, vol. 375, no. 6, pp. 43–50, 2020.
[27] K. Dhrisya, G. Remya and A. Mohan, "Fine-grained entity type classification using GRU with self-attention," *International Journal of Information Technology*, vol. 12, no. 3, pp. 869–878, 2020.
[28] S. Misawa, M. Taniguchi, Y. Miura and T. Ohkuma, "Character-based bidirectional LSTM-CRF with words and characters for Japanese named entity recognition," in *Proc. of the First Workshop on Subword and Character Level Models in NLP*, Copenhagen, Denmark, pp. 97–102, 2017.
[29] D. C. Wintaka, M. A. Bijaksana and I. Asror, "Named-entity recognition on Indonesian tweets using bidirectional LSTM-CRF," *Procedia Computer Science*, vol. 157, pp. 221–228, 2019.
[30] H. Huggard, A. Zhang, E. Zhang and Y. Koh, "Feature importance for biomedical named entity recognition," in *AI 2019: Advances in Artificial Intelligence*, Cham: Springer, pp. 406–417, 2019.
[31] S. Dadas, "Combining neural and knowledge-based approaches to named entity recognition in Polish," in *Artificial Intelligence and Soft Computing*, Cham: Springer, pp. 39–50, 2019.
[32] B. Ji, S. Li, J. Yu, J. Ma, J. Tang *et al.*, "Research on Chinese medical named entity recognition based on collaborative cooperation of multiple neural network models," *Journal of Biomedical Informatics*, vol. 104, no. S3, pp. 103395, 2020.
[33] G. Konoplich, E. Putin, A. Filchenkov and R. Rybka, "Named entity recognition in Russian with word representation learned by a bidirectional language model," in *Artificial Intelligence and Natural Language*, Cham: Springer, pp. 48–58, 2018.
[34] H. Zhao, M. Xu and J. Cao, "Pre-trained language model transfer on Chinese named entity recognition," in *Proc. HPCC/SmartCity/DSS*, Zhangjiajie, China, pp. 2150–2155, 2019.
[35] G. Yu, Y. Yang, X. Wang, H. Zhen, G. He *et al.,* "Adversarial active learning for the identification of medical concepts and annotation inconsistency," *Journal of Biomedical Informatics*, vol. 108, no. 2, pp. 103481, 2020.

[36] H. Xu, Y. Chen, J. Sun, X. Cao and R. Xie, "Iterative strategy for named entity recognition with imperfect annotations," in *Natural Language Processing and Chinese Computing*, Cham: Springer, pp. 512–523, 2020.

[37] Z. Zhang, S. Zhan, H. Zhang and X. Li, "Joint model of entity recognition and relation extraction based on artificial neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3115/1219044, pp. 1–9, 2020.

[38] V. Rudra Murthy and P. Bhattacharyya, "A deep learning solution to named entity recognition," in *Computational Linguistics and Intelligent Text Processing*, Cham: Springer, pp. 427–438, 2018.

[39] D. Zhao, "Named entity recognition based on BiRHN and CRF," in *Green, Pervasive, and Cloud Computing*, Cham: Springer, pp. 465–473, 2019.

[40] W. Zhang, S. Jiang, S. Zhao, K. Hou, Y. Liu *et al.,* "A BERT-BiLSTM-CRF model for Chinese electronic medical records named entity recognition," in *Proc. ICICTA*, Xiangtan, China, pp. 166–169, 2019.

[41] X. Li, H. Zhang and X. H. Zhou, "Chinese clinical named entity recognition with variant neural structures based on BERT methods," *Journal of Biomedical Informatics*, vol. 107, no. 5, pp. 103422, 2020.

[42] Q. Cai, "Research on Chinese naming recognition model based on BERT embedding," in *Proc. ICSESS*, Beijing, China, pp. 1–4, 2019.

[43] S. Yan, J. Chai and L. Wu, "Bidirectional GRU with multi-head attention for Chinese NER," in *Proc. ITOEC*, Chongqing, China, pp. 1160–1164, 2020.

[44] M. Straka, J. Straková and J. Hajič, "Czech text processing with contextual embeddings POS tagging, Lemmatization, Parsing and NER," in *Text, Speech, and Dialogue*, Cham: Springer, pp. 137–150, 2019.

[45] M. Gridach, "Character-aware neural networks for Arabic named entity recognition for social media," in *Proc. WSSANLP2016*, Osaka, Japan, pp. 23–32, 2016.

[46] M. Khalifa and K. Shaalan, "Character convolutions for Arabic named entity recognition with long short-term memory networks," *Computer Speech & Language*, vol. 58, pp. 335–346, 2019.

[47] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 2025, 2019.

[48] M. Gridach and H. Haddad, "Arabic named entity recognition: A bidirectional GRU-CRF approach," in *Computational Linguistics and Intelligent Text Processing*, Cham: Springer, pp.264–275, 2018.

[49] C. Helwe and S. Elbassuoni, "Arabic named entity recognition via deep co-learning," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 197–215, 2019.

[50] M. Ali, G. Tan and A. Hussain, "Bidirectional recurrent neural network approach for Arabic named entity recognition," *Future Internet*, vol. 10, pp. 123, 2018.

[51] M. Ali, G. Tan and A. Hussain, "Boosting Arabic named-entity recognition with multi-attention layer," *IEEE Access*, vol. 7, pp. 46575–46582, 2019.

[52] M. Ali and G. Tan, "Bidirectional encoder-decoder model for Arabic named entity recognition," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9693–9701, 2019.

[53] D. Awad, C. Sabty, M. Elmahdy and S. Abdennadher, "Arabic name entity recognition using deep learning," in *Statistical Language and Speech Processing*, Cham: Springer, pp. 105–116, 2018.

[54] M. Al-Smadi, S. Al-Zboon, Y. Jararweh and P. Juola, "Transfer learning for Arabic named entity recognition with deep neural networks," *IEEE Access*, vol. 8, pp. 37736–37745, 2020.

[55] S. Alzboun, S. 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. SNAMS*, Valencia, Spain, pp. 135–140, 2018.

[56] W. Antoun, F. Baly and H. Hajj, "AraBERT: transformer-based model for Arabic language understanding," in *Proc. of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, Marseille, France, pp. 9–15, 2020.

[57] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," ArXiv: 1609.08144, 2016.

[58] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition," in *Prentice Hall PTR*, 3rd ed. Upper Saddle River: NJ, United States, 2019.

[59] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, Cambridge, Massachusetts, United States: The MIT Press, 2016.

[60] Y. Benajiba, P. Rosso and J. M. BenedíRuiz, "ANERsys: An Arabic named entity recognition system based on maximum entropy," in *Computational Linguistics and Intelligent Text Processing*, Berlin, Heidelberg: Springer, pp. 143–153, 2007.

[61] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. HLT-NAACL 2003*, USA, pp. 142–147, 2003.

[62] B. Mohit, N. Schneider, R. Bhowmick, K. Oflazer and N. A. Smith, "Recall-oriented learning of named entities in Arabic wikipedia," in *Proc. of the 13th Conf. of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, pp. 162–173, 2012.