2019 International Conference on Computer, Control, Electrical and Electronics Engineering (ICCCEEE19)

Proposed Model for Arabic Grammar Error Correction Based on Convolutional Neural Network

Aiman Solyman[1], Zhenyu Wang[1], Qian Tao[1]

[1]School of Software Engineering, South China University of Technology, Guangzhou, China
aiman_mutasem@hotmail.com, wangzy@scut.edu.cn, Taoqian@scut.edu.cn

Abstract—**Deep learning and machine learning algorithms are widely used in Arabic Natural Language Processing (ANLP) aims to develop tools and techniques to process human languages in several forms like written and spoken context. The ANLP still lacks tools and applications to bridge the hole between Arabic and other languages. Furthermore, there are insufficient available resources such as dictionaries, grammatical rules, corpora, etc. Grammatical Error Correction (GEC) one of the NLP tasks seek to develop automatic tools to correct grammar and spelling, the input is incorrect words or sentences and the output become the corrected version of the same sentences. Recently, Neural Networks are used in GEC and had promising results but some improvement is still needed. The limitation of the previous studies are handcrafted and most often extracted from short sentences, also the whole previous Arabic neural approaches are used Recurrent Neural Networks (RNNs). This paper present work-in-progress for developing an Arabic GEC model based on multi convolutional layers with an attention mechanism. Moreover, proposed the incremental techniques and multi-round training model using parallel corpus to get more accuracy and fluently results, also to achieve human-level performance.**

Indexed Terms — **Arabic Natural Language Processing (ANLP), Grammatical Error Correction (GEC), Convolutional Neural Network (CNN).**

I. Introduction

Arabic language classified as a Central Semitic language (CS) [1], and now it is the lingua franca of 22 Arab countries distributed between Africa and Asia [2]. Furthermore, it is one of the six official languages of the United Nations (UN), spoken by about 420 million people and used by one billion six hundred million Muslims in their daily worship [2]. There are two main versions of Arabic language [3], first one is Modern Standard Arabic (MSA), this is the most widely used version in media outlet from TV, movies, newspapers and radio broadcasts. The second version, Classical Arabic (CA) or Quranic Arabic version, it is a little complicated for non-native Arabic speakers. It has special symbols called (*Tanween*-تنوين), used to signify proper pronunciation to give certain effects to words.

Natural language processing (NLP) is a field of computer science that aims to allow users interact with computers using natural languages, in particular, create concepts, find methods, process and analyze large amounts of natural language data [4].

Recently, Arabic Natural Language Processing (ANLP) has received more attention after significant research carried out on other languages like English NLP. Furthermore, several applications have been developed using machine learning (ML) and Deep learning (DL), included text categorization, sentiment analysis, grammar error correction, and dialogue systems [5]. However, ANLP nowadays still lacks the tools to cover various application [4], and the quality of these tools requires additional efforts to bridge the hole between Arabic and other languages. Also, we have to attend to the availability of resources such as dictionaries, grammatical rules, corpora, etc.

Grammatical Error Correction (GEC) is a subfield of NLP, aims to build automatic systems to correct different kinds of errors in text, such as spelling, grammatical, and word choice errors, like human. GEC systems typically work for the sentence correction, taking a potentially erroneous sentence as input and transform it into its corrected version. It also normalized the text from a set of errors, as shown in table 1, listed by [6].

TABLE I
The most famous error list in Arabic language

Error type	Description
Spelling errors	Use an incorrect word with in the sentence
Word Choice Errors	wrong place for the punctuation allocation in the sentence
Punctuation errors	inadequate lexicon and local diallage language
Lexical errors	incorrect derivation or inflection
Morphological errors	grammatical errors, agreement of gender, number
Syntactic errors	Use an incorrect word with in the sentence

Human languages are an irregularity, complexity, and variability of error types, as well as the semantic and syntactic on the context dependencies. Many models were developed to get better performance for text normalization and error correction.

Moreover, the development of some sub-domains in ANLP, like machine translation, question answering, language generation, and multi-document summarization depends on the availability of good grammar models which are covers the entire language. Recently, Neural network achieved promising

978-1-7281-1006-6/19/$31.00 ©2019IEEE

results on GEC using Convolutional Neural networks (CNNs) and Recurrent Neural Networks (RNNs). The power of the CNN is considers feature extraction and classification as one joint task. The input of the most NLP tasks consists of characters, words, sentences or documents represented as a matrix. The available approaches were handcrafted and most often extracted from short sentences. The whole previous Arabic neural approaches for GEC used RNNs. The current project will be the basis for future ALNP projects such as text generation, dialog systems, and semantic parsing. We proposed an encoder-decoder model with nine convolutional layers and attention mechanism, and use machine translation (MT) for Arabic GEC task. The proposed model will be able to apply with other languages like English and Chinese. Moreover, the model will be trained incrementally based on words with rare word segmentation to achieve better and flaunt results.

The rest of this paper is organized as follows: Section II Arabic language challenges. In Section III, Neural network algorithms. The background and related work in Section IV. Section V. the proposed model. The results in Section V1. Finally, the conclusions and future work in the last section.

II. Arabic Language Challenges

The Arabic language has a unique architecture, and it is a little difficulty, and complicity even for native-language speakers. The architecture consists of grammar, spelling, pronunciation, dialogue languages and punctuation marks. The main characteristics and uniqueness of the Arabic language, as follows [7]: 1) right-to-left language means reading and writing from left to right. 2) Arabic language has 28 characters, and some of these characters each one has different shape depends on its position within the word like character / غ gain/ can be used in 4 forms as the following (" غ, "ــع", "ـعـ", "عـ "). 3) The shape of many characters is the same, and we differentiate between them using a dot above or below the letters for example (n-ن, b-ب, t-ت). 4) There are no upper and lower characters, same like Chinese and Korean. 5) Numbers are classified based on gender (feminine and masculine), singular, dual and plural. 6) usually the words comprise several formed roots each root often composed of three letters. 7) designating verbs in the present or future tenses using prefixes but identifying verbs in the past tense by suffixes.

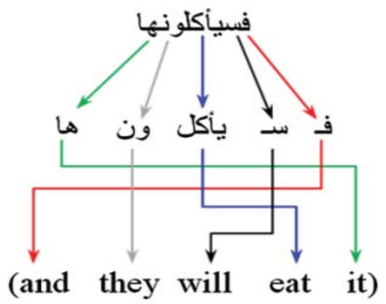

Figure 1. An example for the rich Arabic words morphology and ambiguity [4].

Furthermore, the Arabic language classified as a morphologically rich languages (MRL), that means its words have a more morphological like prefixes and suffixes represent features such as gender, person, number and mood. Moreover, this tends of the MRL languages have more inflect words types [4], this feature increases the degree of stemming and ambiguity from different meanings of the same surface word, figure 1 as an example shows the complexity of MRL for a single word. This ambiguity is aggravated because of some words in the Arabic language has about up to 12 analyses, for an example, the word فسياكلونها *fasayakolonahA* "and thy will eat it" started and ended by a set of suffix and prefix represent for objects, processive or personal.

This kind of complexity and ambiguously effect of ANLP application, for example, the common automatic machine translation software like Google, Bing or Baidu is still far from having a good and high accuracy translation Arabic to other languages [8].

III. Neural Networks Algorithms

The main concept of Artificial Neural Networks (ANN) is trying to simulate the human brain, and the Biological Neural Networks (BNN). The BNN architecture it is an interconnected neurons networks aims to transmit patterns of electrical signals, each node received an input signal and based its output via an axon as an input to another node. Early in 1957, first ANN presented by Rosenblatt it was able to vary its own weights, and have an ability to learn and develop itself to solve linear and simple problems.

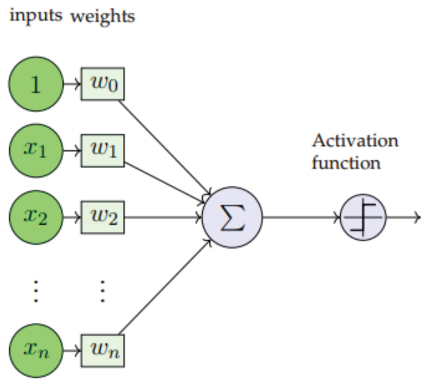

Figure 2. The architecture of a simple Neural Network.

Figure 2 shows the architecture of a simple Neural Network. Moreover, we get the output by computing the linear combination of the input widgets using a nonlinear activation function. Technically, let's consider we have $x_1, x_2, \ldots, x_n$ as input node and $b \in$ bias, and y is the output, calculated by summation of each input widgets $w_1, w_2, \ldots, w_n$ and b, as follow function:

$$y = \sum_{i=1}^{n} w_i x_i + b$$

Recently, the ANN become more complex and the application of NN widely used in our daily life. In this paper, we are focusing on using the ANN in ANLP. There are many algorithms and models, and it has achieved a great success with natural languages processing, as follows:

A. Recurrent Neural Networks

A recurrent neural network (RNN), it is popularly used in NLP, and the main idea is to use cycle connection between each units in the network [9]. This feature worked as embedded memory to keep the network state, used to process sequence of inputs and made the RNN more appropriate for language processing tasks such as speech recognition, dialogue system, and text normalization. The deep RNN has a limitation with the long sequence of text and complex data this problem called vanishing gradient. The vanishing problem makes training parameters less effectively, also more time-consuming and computationally expensive at the earliest layers. To overcome this problem we used Long Short-Term Memory (LSTM) unit [10], aims to categorize data into long term and short term memory cells. This technique allows RNN to determent the important data to be memorized and looped back into the network again, and the rest of the data should be forgotten. Moreover, we used Back-propagation Through Time (BPTT) method [11] fro training RNN.

B. Bidirectional Recurrent Neural Networks

Bidirectional Recurrent Neural Networks (BRNN), invented by Schuster and Paliwal in 1997 [12]. The architecture of BRNN is based on connecting two hidden layers, and passing the input sequence through the opposite side of the two layers. This kind of interconnection allows the output layer to get the information from the previous layer (backward), and the latest layer (forward) simultaneously. BRNN have hidden layers of the two RNN to update and increase the available information of the output. The BRNN are useful when the sequence of the input is need to know the token located before and after the current token. BRNN applications as, grammar and spelling correction, semantic analysis, and handwriting recognition. Training the BRNN is almost similar like RNN training, we used BPTT method with some different because the two hidden layers have no interactions when applying back-propagation, so we need additional processes to update input and output layers, that cannot be done at once. We fix this problem by passing the forward states and backward states first, then network output is passed.

C. Sequence to sequence model

One of the language models used neural networks aims to map a sequence of input words or sentences into the same sequence length output, and different form. Sequence to sequence (seq2seq) or decoder-encoder model [13], has successfully applied in applications such as online chatbots, Google Translate, and voice-enabled devices. The main components of the current model are the encoder and decoder. In the encoder, the process started by receiving the input sequence as $x = (x_1, x_2, \dots , x_n)$, unlike simple BRNN or RNN no need to take into account the state each unit, only keep the last layer state often called context vector or sentence embedding intended of representing the input sentence. On the other hand, passing the context vector and all the previously predicted words $y = (y_1, y_2, \dots, y_n)$ to the decoder. The decoding process predicts the next word y_t. Moreover, the decoder identifying the probability over the output y by eliminating the joint probability. Finally, used the softmax function [14], to normalize the probability distribution for the output of the last layer of the decoder. There are many studies that are used sequence-to-sequence RNN models with LSTM or Attention model to overcome the vanishing problem and improving the quality of the output.

D. Convolutional Neural Network

Convolutional neural network (CNN), it is multilayer perceptron and fully connected networks, that means each node in the layer are connected to all nodes in the next layer [15]. Also, knows as one of the classical neural networks commonly used to analyze and process images. The architecture of CNN consists of the input layer, output layer, and convolutional hidden layers passed results as a mathematical form to successive layers. Recently, CNN applied for NLP tasks like semantic analysis, machine translation, Grammar error correction and achieved promising results. CNN improved the speed of training and computation when we use a large amount of vocabulary.

The current study will use a hybrid of efficient and modern models based on Multi encoder-decoder convolutional layers (nine layers) with an attention mechanism. Also, we will use pre-training word level to each word embedding by initializing the source and target words from a large Arabic corpus using the fasttext tool.

IV. Background and Related work

Machine learning and deep learning showed promising results in automatic spelling and grammar correction. In this section will present the previous methods and techniques for error detection and correction. Generally, the GEC models in the first step used to detect the incorrect spelling words, morphological and syntactic errors in the input text. There are two main techniques used to detect incorrect-word or spelling errors [16], as follows:

A. Dictionary lookup

It is a basic technique aims to compare the input strings with the language resource such as lexicon, dictionary or corpus. The language resource usually contains all inflected forms of the word. Then the system starts looking for the given word, if it did not find in the language resource, will mark it as an un-know word or incorrect word. This kind of techniques improves the search performance and reduce the size of resources through the pattern-matching algorithms and the morphological analysis.

B. N-gram analysis

Used with the statistical models were designed to assign a probability to sequence items such as samples, letters or words in such a sequence order text. Also, for the task of error detection, n-gram analysis used to estimate the likelihood of the given input and accordingly identifying the correct spell word. lately, n-gram analysis is widely used in speech recognition and statistical natural language processing.

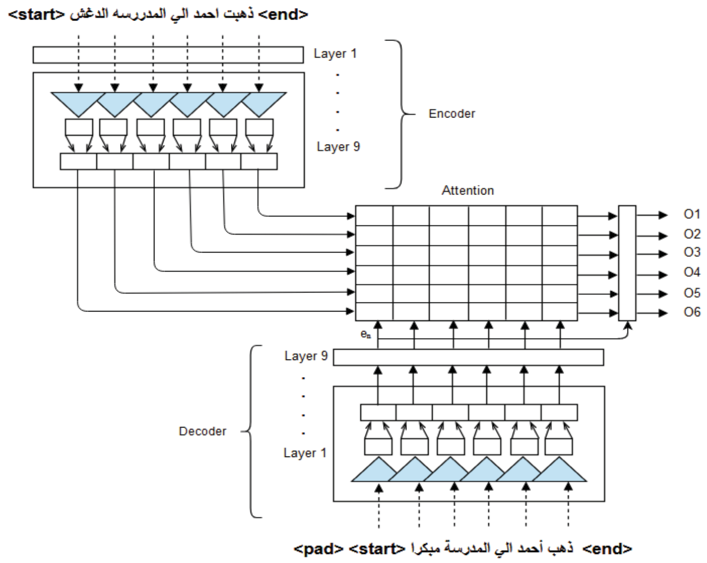

Figure 3. The architecture of the proposed convolutional encoder-decoder model.

Error correction techniques seek to return correct sentences from an incorrect given sentence, the inputs can be considered as the sequence of words and the output are the predicted corrected words, the modern GEC approaches as follows:

A. Rule-based Grammar

Basically, it is about building linguistic rules-based system structures, imitates human which plays a main role in the building and improving these kinds of systems. The biggest advantage of rules-based grammar that is always a way to check the query placed by a user and how it could do that. Also, the whole rules are written by human, controlled and able to report any error to localize and fix by adjusting the rules in the related module. Moreover, the grammar rules-based systems are very flexible manner, and can easily be updated with new data types or functions, and without any significant changes to the rule system. This approach is based on extension of the existing rules, so the system doesn't require a massive training corpus. The limitation the rule-based systems it is requires skilled experts on linguist or a knowledge engineer to manually encode and apply each rule in NLP, and this leads the system development to be more complex. There are many models and approaches for Arabic gramma rule-based such as, [17, 18].

B. Machine-learning and Deep learning algorithms

Recently, it is widely used in GEC task, these algorithms are able to understand human language without being explicitly programmed through statistical methods. Machine learning and Deep learning systems start analyzing the training set to build its own knowledge, produce its own rules and classifiers. Deep learning algorithms are based on probabilistic results, and the obvious advantage of deep learning it's the ability of learnability, also no manual rule grammar coding is needed. Moreover, it is an attractive and simpler alternative is to think of error correction as a translation task. The underlying idea is that a statistical machine translation (SMT) system should be able to translate text written in 'bad' (incorrect) Arabic into 'good' (correct) Arabic, studies used this technique like [19, 20]. There are many studies tried to combining the rule-based and deep learning algorithms into a hybrid system using a grammar-based parser for text-to-SQL translation, and deep learning to complement the rule-based grammar by fixing the syntax, eradicating typos, such studies like [21, 22].

Furthermore, the previous end-to-end Arabic GEC approaches were handcrafted and most often extracted from short sentences, and used recurrent neural networks (RNNs). Moreover, there are limitations on training models based on a limited error correction sentence pairs that not allow models to correct sentences perfectly. Also, using single-round to correct sentences is usually not suitable for sentences with multi grammar errors.

V. Proposed Model

The encoder-decoder models are powerful and widely used in machine translation and text normalization tasks. The encoder network is responsible to map the erroneous source sentence into a fixed length of a context vector. The decoder network seeks to generate the output as a corrected sentence from the context vector. The weakness of the encoder-decoder models is the poor performance with the long input, to address this problem we use attention mechanism [23]. Attention used with the long sequences of text to speed up the training model and allow the decoder network to attend the different sections of the input sentence at each time step when predicting the output sequence.

There are two main limitations for seq2seq models in GEC task [24], as follows: (1) the limited error corrected sentence pairs for training datasets lead to ineffective corrections and fail to achieve totally correct sentence; (2) the seq2seq models usually correct the sentence on single-round inference, and this is not an efficient way to correct complex errors because some corrected errors in a sentence make the context strange. The philosophy of the proposed model is to train the model incrementally by increasing the error-corrected sentence pairs.

Also, generate and use less fluent sentences during training to achieve human-level performance. Generating fluency boost of sentence pairs during training time will be additional training instances at the subsequent training epochs to improve the error correction model performance and correct more sentences. Moreover, to get better results we have to correct the sentence incrementally using fluency boost mechanism with multi-round correction inference [24]. The proposed model will be based on the encoder-decoder architecture using nine convolutional layers with attention mechanism as in [25]. Let's consider we have a sequence S of words as input consists of n tokens as S_1, ... , S_n and $S_i \epsilon V_i$ where V_i refer to the vocabulary source. The vocabulary consists of a number of unique tokens, included a padding token as start-of-sentence and end-of-sentence tokens to make the input sentences have equal length, and the Out of Vocabulary (OOV) token used during inference to replace any character or word outside the training dataset. The current model will use word embedding, each W_i word in the source sentence mapped to row vector C_i initialized and distribution in random uniform between 0 and 1. Then use pre-trained word level for each word embedding by initializing the source and target words from a large Arabic corpus. We will use FastText [18], for word representations which extend to Word2Vec [19] by adding sub-word information to the word vector to overcome the challenge of the small amount of dataset. Also, to deal with the rare words into the parallel Arabic corpus we use a Byte Pair Encoding (BPE) algorithm to split the word into multiple frequent sub-words. This kind of word embeddings calculated by representing a word as a set of characters N-grams and merging the skip-gram embeddings of these characters n-gram sequences using the fastText tool. The architecture of the proposed model consists of nine convolutional encoder and nine decoder layers, as shown in Figure 3.

A. Encoder

The input embeddings are obtained from the previous step which is the pre-trained word level embedding using matrices that are trained before. The first encoder layer is a convolutional filter's response to map every sequence of the consecutive input vectors to a feature vector f. Special tokens are shown in figure3 *<start>* and *<end>* denotes to the beginning and end of the input sentences, and usually used after convolution operations to make sure the returned output vectors as the same number of input tokens number. The output of each convolutional operation will follow by a non-linearity using gated linear units (GLU) [25], where GLU used to reduce vanishing and make the process faster. The input vectors for encoder layer will be added as residual connections layers, and the output vector of the final encoder layer is linearly mapped to get encoder network output vector.

B. Decoder

Now at the decoder, first pad the beginning-of-sentence marker and the previously generated tokens as the same way at the encoder source token. Each embedding linearly mapped passed as input to the first decoder layer, and each decoder layer consists of convolution operations followed by non-linearities GLU, as performed on the previous decoder layers output vectors. The number and size of convolution filters are the same as those in the encoder, also each decoder layer has its own attention module calculated by predicting the target word at each time step t_n plus biases b and the dimension victor weight W multiply by decoder state y as the following equation:

$$z_n^l = W_z Y_N^L + b_z + t_{n-1}$$

The attention weights are computed by using softmax to normalize a dot product of the encoder output vectors e_1, ..., e_n with z_n. The source context vector x_n is computed by summation the encoder output vectors and the source embeddings. The addition of the source embeddings helps to better retain information about the source tokens. The context vector x_n for each layer is linearly mapped of c_n. The output vector of the decoder layer is the summation of x_n, y_n and the previous layers output vector g_n. The final decoder layer output vector g_n is linearly mapped to d_n, where d_n donate to dropout [26], and it is applied before each layer on encoder and decoder network, also embeddings and decoder outputs. Then mapped the decoder output vector to the softmax and target vocabulary size V to computed the target word probabilities. Moreover, to improve the sentences correction and fluency without changing its original meaning will use incremental training to get a fluent sentence. Then, we allow the model to predict n-best outputs S_1, ..., S_n given a correct sentence on the first training round. We also, compare the correctness and fluency of each output to its correct version. If the output sentences fluency score is lower than its correct sentence, we call it a disfluency candidate and we will use the top two results as new input pairs to our model.

The best sequence of target words for Arabic language is obtained by a right-to-left beam search. In a beam search the top d candidates at every decoding time step are retained. The top scoring retrieved candidate at the end of the beam search it will be the correct hypothesis.

C. Dataset

The most appropriate dataset of this project is Qatar Arabic Language Bank[1] (QALB) is created as a part of a collaborative project between Columbia University and the Carnegie Mello University Qatar, funded by the Qatar National Research Fund. The data comes from online commentaries written to Aljazeera articles. The training data contains 2 million words, the development and the test data contain about 50,000 words. The data was annotated and corrected by native Arabic speakers. In our research, we used the release of QALB at ANLPACL 2015 which includes data sets of native and non-native Arabic speakers. QALB corpus is provided in three subsets are: training, development, and test are respectively used as the file extension.

VI. Results

In this stage of the project, we applied the proposed model for the testing set of the QALB corpus, results have been shown in Table 2. Our model without any extra knowledge of NLP and GER processing achieved 40.6 in F0.5. The results will be improved by applying both pre-trained embedding and BPE algorithm as proposed in the previous section. The BPE and Emb algorithms increase the ability to generate unknown word.

TABLE 2
The result of test QALB dataset

P	*R*	*F1*
70.23	72.10	71.14

Due to time suppress, we couldn't apply the whole model architecture to get the final results and this is the primary results. Moreover, we have to increase the amount of the dataset and apply the incremental training to achieve our goal for the task of Arabic GEC.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented the importance and uniqueness of the Arabic language, and also different challenges faced by the ANLP. We have performed a simple survey of the most famous neural network models and algorithms, including RNN, BRNN, and CNN, covered their respective characteristics, advantages, and disadvantages. Then identified which algorithms we would use and why. Furthermore, presented the modern state-of-the-art Grammatical Error Detection and Correction approaches for Arabic and English language. We have also discussed the limitation of the previous studies and the motivation of the current study. Furthermore, we present our proposed model, Arabic GEC based Multi convolutional layers (nine layers), encoder-decoder with an attention mechanism. In addition, we have presented how to implement incremental and multi-round training model using parallel corpus to get more accuracy and fluently results. Then, evaluate our model using QALB 2015 test set to approve our hypothesis against the previous models. To the best of our knowledge, the proposed model will be the first project investigated using incremental training and correcting Arabic grammatical errors based on convolutional neural networks.

REFERENCES

[1] G. Khan, M. P. Streck, and J. C. Watson, The Semitic languages: An international handbook. Walter de Gruyter, 2011.

[2] Istizada. (2019). Complete List of Arabic Speaking Countries Available: http://istizada.com/complete-list-of-arabic-speaking-countries-2014/

[3] K. C. Ryding, A reference grammar of modern standard Arabic. Cambridge university press, 2005.

[4] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, and I. Abunadi, "Arabic natural language processing and machine learning-based systems," IEEE Access, vol. 7, pp. 7011-7020, 2019.

[5] A. A. Al-Ajlan, H. S. Al-Khalifa, and A. S. Al-Salman, "Towards the development of an automatic readability measurements for Arabic language," in 2008 Third International Conference on Digital Information Management, 2008, pp. 506-511: IEEE.

[6] W. Zaghouani et al., "Large scale arabic error annotation: Guidelines and framework," 2014.

[7] H. Hasanuzzaman, "Arabic language: characteristics and importance," The Echo. A Journal of Humanities & Social Science, vol. 1, no. 3, pp. 11-16, 2013.

[8] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT, 2003, pp. 1-8: Association for Computational Linguistics.

[9] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," arXiv preprint arXiv:1312.6026, 2013.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[11] P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol. 78, no. 10, pp. 1550-1560, 1990.

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104-3112.

[14] M. Costa, "Probabilistic interpretation of feedforward network outputs, with relationships to statistical prediction of ordinal quantities," International journal of neural systems, vol. 7, no. 05, pp. 627-637, 1996.

[15] Y. Jia et al., "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675-678: ACM.

[16] K. Kukich, "Techniques for automatically correcting words in text," Acm Computing Surveys (CSUR), vol. 24, no. 4, pp. 377-439, 1992.

[17] K. Shaalan, "Rule-based approach in Arabic natural language processing," The International Journal on Information and Communication Technologies (IJICT), vol. 3, no. 3, pp. 11-19, 2010.

[18] K. F. Shaalan, "Arabic GramCheck: A grammar checker for Arabic," Software: Practice and Experience, vol. 35, no. 7, pp. 643-665, 2005.

[19] D. Watson, N. Zalmout, and N. Habash, "Utilizing Character and Word Embeddings for Text Normalization with Sequence-to-Sequence Models," arXiv preprint arXiv:1809.01534, 2018.

[20] S. Ahmadi, "Attention-based encoder-decoder networks for spelling and grammatical error correction," arXiv preprint arXiv:1810.00660, 2018.

[21] N. Zalmout and N. Habash, "Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 704-713.

[22] M. Nawar, "CUFE $@ $ QALB-2015 Shared Task: Arabic Error Correction System," in Proceedings of the Second Workshop on Arabic Natural Language Processing, 2015, pp. 133-137.

[23] A. Vaswani et al., "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998-6008.

[24] T. Ge, F. Wei, and M. Zhou, "Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study," arXiv preprint arXiv:1807.01270, 2018.

[25] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 1243-1252: JMLR. org.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.