Journal of Theoretical and Applied Information Technology
31st August 2020. Vol.98. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEEP LEARNING BASED ABSTRACTIVE ARABIC TEXT SUMMARIZATION USING TWO LAYERS ENCODER AND ONE LAYER DECODER

[1]DIMA SULEIMAN AND [2]ARAFAT AWAJAN

[1]The University of Jordan, Amman, Jordan

[1]Princess Sumaya University for Technology, Amman, Jordan

[2]Princess Sumaya University for Technology, Amman, Jordan

E-mail: [1]d.suleiman@psut.edu.jo, [2]Awajan@ psut.edu.jo

ABSTRACT

In this paper, an abstractive Arabic text summarization model is proposed, which is based on sequence-to-sequence recurrent neural network encoder decoder architecture. The proposed model consists of two layers of hidden states at the encoder and one layer of hidden states at the decoder. The encoder and decoder layers use long short-term memory. The two layers of the encoder are the input text layer and the name entities layer. The inputs for the input text layer are the word embedding of the input text words, while the inputs for the name entity layer are the word embedding of the input text name entities. In all layers, the word embedding that is used is one of the AraVec pre-trained word embedding models. Furthermore, global attention mechanism is used by the decoder to generate the summary words. Special dataset is collected and used for training and evaluating the abstractive summarization model. Moreover, the proposed model is evaluated using ROUGE1 and ROUGE1-NOORDER evaluation measures. The experimental results show that, the proposed model provides good results in terms of ROUGE1 and ROUGE1-NOORDER where the values are 38.4 and 46.4 respectively. Finally, a comparison is made between the word2Vec and dependency parsing based word2Vec word embedding models. The abstractive summarization models that use dependency based word2Vec model outperformed the models that use the original word2Vec model. As a result, the quality of the word embedding highly affects the quality of the generated summary.

Keywords: *Deep Learning, Abstractive text summarization, Recurrent Neural Network, Attention Mechanism, LSTM, ROUGE.*

1. INTRODUCTION

Recently, huge amount of Arabic text such as articles, documents and news need to be summarized automatically in order to be useful [1]. The importance of automatic summarization refers to several reasons including extraction and loading of crucial information efficiently and rapidly in reasonable time [2], [3]. Text summarization can be classified, based on the nature of the generated summary, into two types: extractive and abstractive summaries. Extractive summarization extracts the summary, which is related to significant parts of the text, based on statistics and linguist features, while abstractive summarization based on the real semantics of the text. Therefore, in extractive summarization, the summary is literally appears in the text while the abstractive summary may never appear in the text [4].

On the other hand, extractive summarization is easier than abstractive one, since abstractive summarization requires semantic analysis of the text that can be achieved using advanced natural language processing (NLP) and machine learning techniques [5], [6]. However, abstractive summarization is better, since the abstractive summary is more meaningful because it is like the human generated summary [7]. In both extractive and abstractive summarization, the summary must

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

have common characteristics such as: the meaning of the summary must be the same as the meaning of the text, the sequence of the sentences in the summary must be the same as the sequence of the sentences in the text, the summary length must be less than the text length, and finally the repletion of the words in the summary must be minimized [2], [8].

In this paper, the first contribution is to propose an Arabic abstractive text summarization model that is based on deep learning. In English abstractive summarization, deep learning was used for the first time by Rush et al. in 2015 [9]. However, to the best of our knowledge, there are no published research that use deep learning in Arabic abstractive text summarization. Abstractive text summarization models that utilized deep learning techniques provided significant results in the recent year. Therefore, the proposed model is based on Recurrent Neural Network (RNN) encoder decoder model [10]. The architecture of the proposed model consists of two layers of bidirectional Long Short-Term Memory (LSTM) at the encoder and one layer of unidirectional LSTM at the decoder. Furthermore, global attention mechanism was used at the decoder. The input of the first encoder layer is the word embedding of the text while the input of the second encoder layer is the word embedding of the input text name entities. On the other hand, the input for the decoder is the word embedding of the summary words. The word embedding that is used is one of the pre-trained AraVec word embedding models. [11]. The experiments of the proposed abstractive summarization model are performed in a dedicated dataset that we collected specially for this purpose. Finally, the quality of the generated summary is evaluated using ROUGE1 and ROUGE1-NOORDER evaluation measures. Another contribution of this paper is to study the effect of the quality of the word embedding on the quality of the generated summary. Therefore, a comparison has been made between the original word2Vec model and dependency based worde2Vec word embedding model.

The rest of this paper is organized as follows: section 2 covers the related work, while the details of the proposed model are explained in section 3. Section 4 discusses the experimental results and evaluation. The effect of the word embedding quality is studied in section 5. Finally, the conclusion is covered in section 6.

2. RELATED WORK

Deep learning was used in English abstractive text summarization for the first time in 2015 by Rush et al.[9]. However, to the best of our knowledge it has not yet been used in Arabic. Rush et al. proposed to use three types of encoder including attention, convolution and bag of words encoders. Moreover, the local attention mechanism was used by the decoder which conditions every output word to the input words within context window. On the other hand, beam search was used to select the best target words. Training was conducted using Gigaword dataset while testing was performed using DUC-2003 and DUC-2004 datasets. Datasets were subjected to several pre-processing stages included using lower case letter, UNK token to represent the least frequently words, tokenization, and using the symbol # to replace all digits. ROUGE1, ROUGE2 and ROUGE-L was used for evaluating the quality of the generated summary where the value of ROUGE1 was 28.18 while the values of ROUGE2 and ROUGE-L were 8.49, 23.81 respectively.

RAS (Recurrent Attentive Summarizer) Abstractive sentence summarization method which is an extension of Rush et al. method was proposed in [12]. But instead of using feed forward neural network, RAS used recurrent neural network (RNN). The summarization model was trained using Gigaword dataset, while the model was evaluated using DUC-2004 dataset. Moreover, the same pre-processing stages that were conducted by [9] were also conducted by RAS. The summary was evaluated using ROUGE1, ROUGE2 and ROUGE-L with values 28.97, 8.26 and 24.06 respectively. After this research, most of the abstractive summarization research used RNN sequence-sequence encoder decoder model.

Nallapati et al. used attention mechanism with RNN encoder decoder architecture to propose an abstractive text summarization model [13]. The encoder of the model consists of two layers of the bidirectional GRU-RNN. One layer represents the word level while the second layer is used for the sentence level. On the other hand, the decoder layer consists of unidirectional GRU-RNN where the softmax was used to generate the summary words. Several features of the input text such as the name entities, part of speech tagging and TF-IDF in addition to the word embedding of the words are fed to the encoder. Word2vec was used to convert the words into vectors. Furthermore, the training of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the model was conducted in DUC, Gigaword and CNN/Daily datasets. The generated summaries were evaluated using ROUGE1, ROUGE2 and ROUGE-L where the values were 35.46, 13.3 and 32.65 respectively.

RNN sequence-sequence encoder decoder model that utilizes LSTM was proposed in [29]. This model used three global attention mechanisms including dot product, bilinear and scalar value. Three architectures were proposed, the first one consists of unidirectional LSTM in both encoder and decoder. The second one consists of a bidirectional LSTM encoder and unidirectional LSTM decoder. The last one consists of bidirectional LSTM in both encoder and decoder. Moreover, Glove word embedding was used in the second and third architecture while in first architecture the embedding was generated during the training of the model. All models were evaluated using BLEU evaluation measure.

Zhou et al. proposed abstractive summarization model which consists of encoder, decoder and selective gate, this model is called Selective Encoding for Abstractive Sentence Summarization (SEASS) model [14]. SEASS model consists of bidirectional GRU encoder and unidirectional GRU decoder. The representation of the words of the sentences was generated using the selective gate. Several datasets were used for training and testing the model such as DUC 2004, Gigaword and MSR-ATC datasets. Datasets were pre-processed using several stages such as normalization, using hunki to replace the less frequent words, using the symbol # to replace all the digits and tokenization. Furthermore, the best target word was selected using the beam search. Finally, the model was evaluated using ROUGE1 where its value was 36.15, ROUGE2 with 17.54 value and ROUGE-L with the value of 33.63.

A Dual attention model which consists of two encoders with bidirectional GRU and one decoder that has a gate network of dual attention was proposed for abstractive text summarization [15]. In their model, instead of generating one context vector as in all previous models, two context vectors are used and merged by the decoder. The experiments were conducted using Gigaword dataset that was processed using the same steps that were used in [9]. ROUGE1, ROUGE2 and ROUGE-L were used for evaluating the quality of the generated summary with values 37.27, 17.65, and 34.24 respectively.

In all previous research, the target summary was single-sentence summary. However in all the following research the target summary is multi-sentence summary. Single layer bidirectional LSTM encoder and single layer unidirectional LSTM decoder with an attention mechanism was used by See et al. to propose multi-sentence summary abstractive summarization model [16]. See et al. model proposed solutions for several abstractive summarization challenges such the repetition of words in the generated summary and inaccurate information in some cases. CNN/Daily Mail dataset was used for training and testing the model. In addition, the word embedding of the input text was pre-trained. Even more, ROUGE1was used for evaluating the summary in addition to using ROUGE2 and ROUGE-L where the values were 39.53, 17.28 and 36.38 respectively.

Adversarial process was used to propose an abstractive summarization model, which consists of generative and discriminative models [17]. In the first step, the generated summary is optimized using the reinforcement learning. In the next step, the generated summary is classified, using the discriminator model, into either ground truth or machine generated summaries. The architecture of the adversarial process consists of an encoder with bidirectional LSTM and decoder with unidirectional LSTM that utilizes an attention mechanism.
CNN/Daily mail dataset was used in experiments of the model that was evaluated using ROUGE1 with value 39.92, ROUGE2 and ROUGE-L with values 17.65, 36.71 respectively. Finally, the generated summary was also evaluated by human by choosing 50 test examples randomly from the dataset.

The semantic phrases of the input text were considered when generating abstractive summary using LSTM-CNN model called ATSDL [18]. ATSDL was trained and evaluated using CNN/Daily mail dataset. The model was evaluated using ROUGE1 where its value was 34.9 and ROUGE2 with value 17.8.

Both abstractive and extractive methods were used in guiding generation model to generate abstractive summary [19]. Guiding generation model architecture composed of an encoder with bidirectional LSTM and decoder with a unidirectional LSTM decoder. Several techniques were utilized including attention mechanism and pointer network in addition to using softmax layer.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

At the encoder, the keywords of the text were encoded using Key Information Guide Network (KIGN). Finally, CNN / Daily Mail dataset was used in experiments where ROUGE1, ROUGE2 and ROUGE-L were used for evaluating the models where their values were 38.95, 17.12 and 35.68 respectively.

Yao et al. proposed a (DEATS) model to generate abstractive summary where the model consists of dual encoding and one level of GRU at the decoder, in addition to utilizing attention mechanism [20]. The dual encoding consists of primary and secondary levels of encoding where the primary level is the same as the standard encoder level. On the other hand, the secondary level generates new context vector based on the previous input and output. CNN/Daily Mail and DUC 2004 datasets were used in entire experiments. Furthermore, ROUGE1, ROUGE2 and ROUGE-L were used for evaluating DEATS model where their values were 40.85, 18.08 and 37.13 respectively.

3. PROPOSED METHOD

The proposed abstractive Arabic text summarization model is based on LSTM RNN encoder decoder model. In this section, the three stages of the proposed model including the collection of the dataset, the architecture of the proposed model and the evaluation metric used for evaluating the quality of the generated summary are discussed.

3.1 Dataset Collection

DUC-2004 [21] and Gigaword [22] are two datasets that were used in Arabic abstractive text summarization. There are several problems associated with these datasets. The first problem is the size of DU2004 dataset which is small. The second problem is that, the summary in DUC2004 is written in English instead of Arabic while the text is written in Arabic. On the other hand, while the Gigaword dataset is large, it is not free. Accordingly, in this research, a dedicated dataset is collected from several resources such as Reuters, Aljazeera and others. The dataset consists of documents where each document must have a text and single line summary. The total number of documents of our dataset before pre-processing is 79965 documents where around 69024 documents are taken from SANAD_SUBSET [23] and 10932 documents are gathered from news websites.

3.2 Multi-layer Encoder Model Architecture

In recent years, sequence-to-sequence architecture is highly used in NLP applications that have a sequence of inputs and sequence of outputs such as machine translation and text summarization. In this research, encoder decoder sequence-to-sequence architecture that consists of LSTM RNN is used. The baseline architecture of the proposed model is the same as the architecture that was used by Nallapati et al. [13]. The proposed model consists of two-layer encoder and one-layer decoder. The two hidden states layers at the encoder are the input text layer and the name entities of the input text layer. In both layers, bidirectional LSTM is used. On the other hand, there is one layer of hidden states at the decoder which consists of unidirectional LSTM.

The input of the input text layer is the word embedding of the input text words. Also, the input for the name entity layer is the word embedding of the input text name entities and finally, the word embedding of the summary words are the inputs for the hidden states at the decoder layer. Figure 1 shows the architecture of the proposed model. The input text layer passes the representation of the sequence of the input text $x=\{x_1, x_2, x_3, \ldots, x_n\}$ to the hidden states $h=\{h1, h2, h3, \ldots, hn\}$. Moreover, the representation of the sequence of the name entities $nne=\{ne1, ne2, ne3, \ldots, nee\}$ at the second layer is mapped to the second layer hidden states $hne=\{ hne1, hne2, hne3, \ldots, hnee \}$. The last backward hidden state and the last forward hidden state are concatenated to generate the name entities representation *nev*.

At the decoder, there is a difference between the training phase and the testing phase in term of the input for the decoder hidden states. During training, the inputs for the hidden state *st* at time step *t* are the output of the previous hidden state along the side with the word embedding of the next word in reference summary, where the reference summary is the target summary. On the other hand, during testing, the inputs for the hidden state *st* are the output of the previous hidden state along the side with the word embedding of the previously generated summary word, since there is no reference summary. Furthermore, the inputs for the first hidden state in both training and testing are the word embedding of the symbol <SOS> and the context vector.

Journal of Theoretical and Applied Information Technology
31st August 2020. Vol.98. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

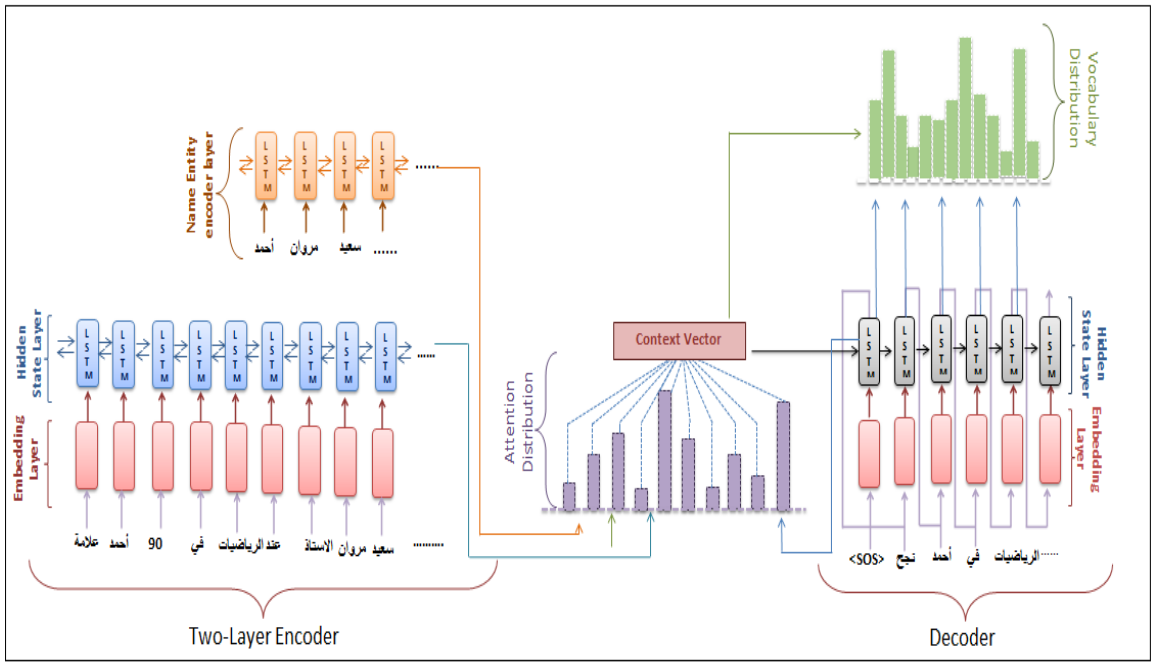

Figure 1: Two-layer encoder abstractive Arabic text summarization

The context vector is the vector representation of the encoder hidden states whose size is fixed. Attention mechanism is used to generate the context vector by calculating the weight between all the input words and each output word [9]. Attention mechanism provides a focus of the input part that has high significant effect on the output. There are several types of attention mechanism such as local and global attention. The proposed model considered global attention mechanism instead of local since it provided better results. The target word y_t is predicted using the context vector c_t which is computed using equations 1, 2 and 3.

$$etl = v^t * \tanh(W_h hl + W_{ne} nev + W_s st) \quad (1)$$

$$a_t = \text{softmax}(etl) \quad (2)$$

$$c_t = \sum_{l=1}^{n} a_{tl} h_l \quad (3)$$

Where v^t, W_h, W_{ne}, W_s are learnable parameters, a_t is the attention distribution (probability distribution over the source text) and h_i is the hidden state of the input sequence *xi*. The softmax layer at the decoder is used to predict summary words y_t based on the vocabulary distribution. The input for the softmax layer is the concatenation of three components including the hidden state of the decoder s_t, the vector representation of the name entities *nev* and the context vector c_t as shown in Equation 4, where $P(y_t|y_1,y_2,\ldots, y_{t-1})$ is the probability of the current target summary word given the previous predicted words. In testing, greedy search is used to pick the predicted word over the probability distribution.

$$p(y_t|y_1,\ldots,y_{t-1}) = \text{softmax}(f([s_t, hv, nev, c_t])) \quad (4)$$

3.3 Abstractive Summary Evaluation

The most common evaluation measure that is used for evaluating the quality of the generated summary in both abstractive and extractive summarization is called ROUGE. ROUGE is a package that stands for (Recall-Oriented Understudy for Gisting Evaluation) [24]. There are several evaluation measures in ROUGE such as ROUGE-N and ROUGE-L. With all the measures a comparison is made between the generated summary and the reference summary. ROUGE-N is calculated using Equation 5. In ROUGE-N, *N* represents the *N* gram, therefore ROUGE1 represents unigram and ROUGE2 represents bigram.

$$ROUGE-N = \frac{\sum_{s \in \{ReferenceSummaries\}} \sum_{gramn} Countmatch(gramn)}{\sum_{s \in \{ReferenceSummaries\}} \sum_{gramn} Count(gramn)} \quad (5)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Where N is the n-gram length, S is the reference or target summary and Count(gramn) is the number of n-gram words the appear in the reference summary. Finally, Countmatch(gramn) is the maximum number of n-gram match between the reference summary and the target summary. In this research ROUGE1 is used for evaluating the quality of the summary. We will use ROUGE1 in two ways, in the first way we will consider the order of the words in the reference summary, while in the second way; we will discard the order of the words. We will use the name ROUGE1 for the first way. Also, we will use the name ROUGE1-NOODER for the second way. For example, if we have the reference summary R and the generated summary G then:

R: Sami read a book
G: the book read by Sami

By using ROUGE1 the match is between the word Sami in reference summary and the word Sami in the generated summary only while by using ROUGE1-NOORDER the match occurs between three words: Sami, read and book.

4. EXPERIMENTAL RESULTS AND EVALUATION

4.1 Dataset Pre-processing

The collected dataset is subjected to several pre-processing stages include removing of the foreign languages, diacritic and punctuation marks. In addition, in order to minimize the training time, the summary and input text length is restricted to 15 and 600 tokens respectively. As a result, the number of the documents in the datasets is minimized to 61824 documents. Since the input of the encoder and decoder is the word embedding of the words, in this research, we used several word embeddings. The first word embedding that we used is one of the pre-trained AraVec word embedding models [11]. The other word embeddings are the word2vec based models that we trained over OSAC corpus. In this section, the focus is on the first one. However, more details about the other word embeddings are covered in the next section. AraVec is an Arabic open source project that presents several word embedding models, which were trained over 3,300,000,000 tokens collected from several resources such as Wikipedia Arabic articles and Twitter. In this paper, we selected one of the AraVec models that was trained over 1800000 Wikipedia documents. The dimension size of the vectors that represent the words in the selected model is 300. The same word embedding model is used for representing the input text words, name entities and the summary words. In some cases, some words did not have any embedding; therefore in this case, we proposed to use the embedding of their stem. For example, if the word "أبواب"which means doors is not found in the word embedding vocabulary, then the word embedding of their stem "باب" (door) is used instead.

Furthermore, the second encoder layer of the proposed model is the name entities of the input text layer. Therefore, the name entities must be recognized before training. In this paper, the name entities are recognized using Farasa name entity recognizer [25].

4.2 Experimental Settings

The total number of documents that is used in experiments is 61824 documents. The documents are partitioned into three parts including training, validation and testing. There are 49984, 8006 and 3834 documents in training, validation and testing respectively. Furthermore, the vocabulary size of both the source and the target is 219758 tokens and the dimension size of the vector that represents each word is 300. In addition, the number of hidden sates of the input text layer and the decoder layer is 600 for each. On the other hand, the number of hidden states in the name entities layer is 10. Finally, the number of epochs in each experiment is 50 epochs. Also, batching is used to minimize the training time where the batch size is 64. All of the experiments are performed on Standalone Workstation running under Windows Server 2016 Data centre with Intel Xeon Silver 4114 2.20HHz Nvidia Quadro P5000 and 64GB RAM using Nvidia CUDA 9.0, Python 3.6 and TensorFlow 1.12.0.

Several variations of the proposed model are implemented in order to make a comparison, since there are no published research on deep learning based abstractive Arabic text summarization. In all variations bidirectional LSTM is used at the encoder and unidirectional LSTM is used at the decoder. The following are the models.

1) Original_rnn_lstm: global attention mechanism is used with the dropout value of 0.3.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

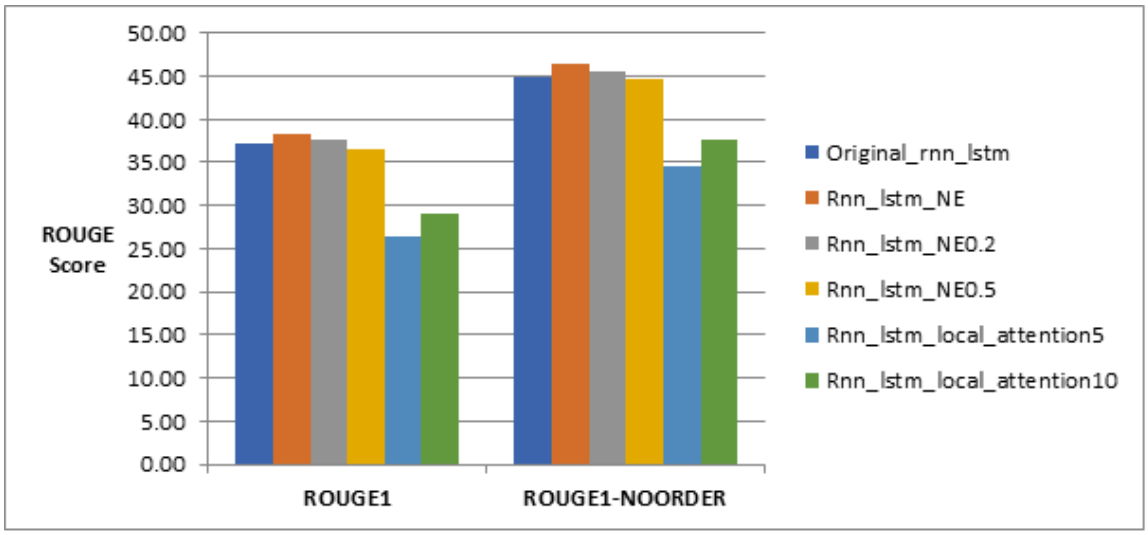

Figure 2: Performance comparisons of various abstractive Arabic text summarization models

2) Rnn_lstm_NE: this model is the proposed model.
3) Rnn_lstm_NE0.2: this model is the proposed model with the dropout value of 0.2.
4) Rnn_lstm_NE0.5: this model is the proposed model with dropout value of 0.5.
5) Rnn_lstm_local_attention5: this model is the same as the first model original_rnn_lstm but instead of using global attention, local attention is used as proposed in [26] where the window size is 5.
6) Rnn_lstm_local_attention10: this model is the same rnn_lstm_local_attention5 but the window size is 10.

Table 1: Performance comparisons of various abstractive Arabic text summarization models

Model	ROUGE1	ROUGE1-NOORDER
Original_rnn_lstm	37.3	45.0
Rnn_lstm_NE	38.4	46.4
Rnn_lstm_NE0.2	37.7	45.6
Rnn_lstm_NE0.5	36.6	44.6
Rnn_lstm_local_attention5	26.4	34.5
Rnn_lstm_local_attention10	29.1	37.7

4.3 Results and Discussions

The proposed abstractive summarization model and all its variations are trained, validated and tested over the dataset that was described in previous sections. The results of testing the models are displayed in Table 1 and Figure 2. As shown, the best results in term of ROUGE1 and ROUGE1-NOORDER are achieved by the proposed model Rnn_lstm_NE with values 38.4 and 46.4 respectively. It seems that, the multi-layer encoder approaches provided better results than the single-layer encoder. Adding more layers at the encoder provides additional information about the text that helps in improving its representing. As a result, the prediction of the summary words becomes better.

Moreover, one of the proposed model parameters which is the dropout parameter is tuned using three values which are 0.2, 0.3 and 0.5. The entire experiments show that, the best results are achieved with dropout value equal to 0.3 which is the value of the proposed model as in Rnn_lstm_NE. All models used global attention except the last two models where local attention is used. In Rnn_lstm_local_attention5, the window size is 5 while in Rnn_lstm_local_attention10, the window size is 10. Based on the results of evaluating the models, using global attention is better than using local attention. Moreover, with local attention, the results become better when the window size is large. Therefore, the results of Rnn_lstm_local_attention10 model are better than the results of Rnn_lstm_local_attention5 model. It can be clearly seen that, in all the models, the values of ROUGE1-NOORDER are always greater than the values of ROUGE1 since ROUGE1-NOORDER has less restrictions.

As a result, the multi-layer encoder model, which consists of input text layer and name entity layer, provides the best results where the name entity layer improved the representation of the input text.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. WORD EMBEDDING QUALITY

Word embedding is the representation of the words using vectors [27]. As mention before, the word embedding of the words is highly used in NLP application especially when dealing with neural network [28]–[31]. The input for neural network is the word embedding of the words instead the words themselves. In the proposed abstractive Arabic text summarization model, the word embedding is used several times including the words of the input text, the name entities and the words of the summary. In this section, we study the effect of the quality of the word embedding on the quality of the generated summary. We compared between two word embeddings models: the first one is the original skip-gram word2Vec model while the second one is the dependency based skip-gram word2Vec model, more details are covered in the following subsection.

5.1 Skip-gram Word2Vec Model

Word2vec is a neural network that consists of input, hidden and output layers. At the hidden layer, there is no activation function. The number of neurons in this layer is equal to the size of the dimension of the vector that represents each word. Softmax function is the objective function at the output layer. Word2vec has several hyper parameters such as the vocabulary size and the context window. The vocabulary size is number of the most frequent words in the corpus where the model must be trained on large corpus. On the other hand, the context window is the number of words that must be considered when calculating the probability of word/words given other word/words. There are two approaches of word2Vec include continuous bag of words (CBOW) and skip-gram. The focus of this section is on skip-gram approach. In skip-gram approach, the input for the model is the middle word while the output is the sorrowing words within the context window size. For example if the window size is 7 then the number of the words at the output is 6 which are the three words to the left of the middle word and three words to the right of the middle word. In this case, the objective is to find the probability of the middle word given the surrounding six words.

5.2 Dependency Based Skip-Gram word2Vec Model

The difference between the original skip-gram model and dependency based skip-gram model is the size of the context window. In dependency based model, dependency parsing is used to determine the relationship between words. Therefore, the context window size will increase to include all the words that have a relationship with the middle word. In this case, the context window size becomes dynamic. As a result, the probability of the occurrence of the word by considering the neighbors and dependent words is more accurate.

5.3 Dataset Pre-processing

In this section, OSAC corpus is used to train both word embedding models [32]. OSAC consists of 22,429 documents and subjected to several pre-processing stages such as removing foreign languages, diacritical mark and punctuation marks. In this section, we have not used AraVec model since the dataset that was used in the training AraVec model is not available. Therefore, in order to make a common sense comparison, the same dataset must be used to train the two word embedding models. Finally, Arabic Stanford dependency parsing is used to determine the dependency parsing of the sentences [33].

5.4 Experimental Settings

The two models are trained on the same standalone computer running under Windows 10 with 3.4 GHz Intel Core i7 quad processor and 24 GB RAM. Moreover the version of TensorFlow and python are v1.12.0 and v3.5.3 respectively. The same hyper parameters are used in both models with the following values, the context window size is 7 and the dimension size is 100. Also, the size of the vocabulary is 50000.

5.5 Results and Discussions

The results of using the two word embedding models in text summarization can be seen in Table 2. In all the variation models of text summarization, we can notice that the dependency based skip-gram model provides better results than the original skip-gram model in terms of ROUGE1 and ROUGE1-NOORDER. The reason for this is that, in dependency based model, the context window

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

become larger and considers far away words that have a relationship with the middle word. Also, if a comparison is made in term of ROUGE1 and ROUGE1-NOORDER values between the original skip gram model and the AraVec model, the results show that, the training using AraVec model provides better result. This referred to the fact that, the AraVec models are trained over a very large corpus. Accordingly, the quality of the word embedding model highly affects the quality of the generated summary. Table3 shows an example of the text, ground truth summary and generated summary using the proposed model Rnn_lstm_NE.

6. CONCLUSION

In this paper, an abstractive Arabic text summarization model was proposed. The proposed approach architecture consists of two layers at the encoder and one layer at the decoder. Both layers at the encoder utilized bidirectional LSTM while the decoder layer utilized unidirectional LSTM. Moreover, the decoder used global attention mechanism. The two layers at the encoder are the input text layer and the name entities of the input text layer. The input for the input layer is the word embedding of the input text words while the input for the name entity layer is the word embedding of the name entities. The experiments were conducted in a dataset that was collected and pre-processed to be suitable for the abstractive summarization. Even more, the quality of the proposed model was evaluated using ROUGE1 and ROUGE1-NOORDER evaluation measure. ROUGE1 and ROUGE1-NOORDER measure the similarity between the words of the generated summary and the words of the reference summary.

The results showed that, the proposed model outperformed the other models. Using multi-layer encoder add more features to the input text which improved its representation. The quality of the representation of the text highly affects the quality of the generated summary.

Finally, several word embedding models were used to study the effect of the quality of the word embedding on the quality of the generated summary. The entire results showed that, the quality of the summary is highly affected by the size of the corpus that is used to train the word embedding model. Also, the dependency based skip-gram model generated better summary quality than the original skip-gram model since the quality of the dependency based word embedding model is high.

Acknowledgment: This research was supported by Scientific Research Support Fund, Jordan.

Table 2: Performance comparisons of various abstractive Arabic text summarization models using several word embedding models

Model		Original skip-gram	Dependency based Skip-gram	
	ROUGE1	ROUGE1-NOORDER	ROUGE1	ROUGE1-NOORDER
Original_rnn_lstm	9.5	15	15.5	22.7
Rnn_lstm_NE	8.8	14.7	14.7	21.2
Rnn_lstm_local_attention5	7.8	15.9	4	7
Rnn_lstm_local_attention10	8	15.3	13.1	20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] M. Allahyari et al., "Text Summarization Techniques: A Brief Survey," International Journal of Advanced Computer Science and Applications, vol. 8, no. 10, 2017, doi: 10.14569/IJACSA.2017.081052.

[2] A. B. Al-Saleh and M. E. B. Menai, "Automatic Arabic text summarization: a survey," Artificial Intelligence Review, vol. 45, no. 2, pp. 203–234, Feb. 2016, doi: 10.1007/s10462-015-9442-x.

[3] H. A. Ghafoor, A. Javed, A. Irtaza, H. Dawood, H. Dawood, and A. Banjar, "Egocentric Video Summarization Based on People Interaction Using Deep Learning," Mathematical Problems in Engineering, vol. 2018, pp. 1–12, Nov. 2018, doi: 10.1155/2018/7586417.

[4] D. Suleiman and A. A. Awajan, "Deep Learning Based Extractive Text Summarization: Approaches, Datasets and Evaluation Measures," in 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, Oct. 2019, pp. 204–210, doi: 10.1109/SNAMS.2019.8931813.

[5] A. M. Azmi and N. I. Altmami, "An abstractive Arabic text summarizer with user controlled granularity," Information Processing & Management, vol. 54, no. 6, pp. 903–921, Nov. 2018, doi: 10.1016/j.ipm.2018.06.002.

[6] A. Elrefaiy, A. R. Abas, and I. Elhenawy, "Review of recent techniques for extractive text summarization," Journal of Theoretical and Applied Information Technology, . Vol., no. 23, p. 21, 2005.

[7] C. Sunitha, A. Jaya, and A. Ganesh, "A Study on Abstractive Summarization Techniques in Indian Languages," Procedia Computer Science, vol. 87, pp. 25–31, 2016, doi: 10.1016/j.procs.2016.05.121.

[8] A. M. Azmi and S. Al-Thanyyan, "A text summarizer for Arabic," Computer Speech & Language, vol. 26, no. 4, pp. 260–273, Aug. 2012, doi: 10.1016/j.csl.2012.01.002.

[9] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," arXiv:1509.00685 [cs], Sep. 2015, Accessed: Aug. 02, 2018. [Online]. Available: http://arxiv.org/abs/1509.00685.

[10] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.

[11] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," Procedia Computer Science, vol. 117, pp. 256–265, 2017, doi: 10.1016/j.procs.2017.10.117.

[12] S. Chopra, M. Auli, and A. M. Rush, "Abstractive Sentence Summarization with Attentive Recurrent Neural Networks," 2016, pp. 93–98, doi: 10.18653/v1/N16-1012.

[13] R. Nallapati, B. Zhou, C. N. dos santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," arXiv:1602.06023 [cs], Feb. 2016, Accessed: Aug. 02, 2018. [Online]. Available: http://arxiv.org/abs/1602.06023.

[14] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective Encoding for Abstractive Sentence Summarization," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1095–1104, 2017, doi: 10.18653/v1/P17-1101.

[15] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the Original: Fact Aware Neural Abstractive Summarization," arXiv:1711.04434 [cs], Nov. 2017, Accessed: Jun. 29, 2019. [Online]. Available: http://arxiv.org/abs/1711.04434.

[16] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," arXiv:1704.04368 [cs], Apr. 2017, Accessed: Aug. 02, 2018. [Online]. Available: http://arxiv.org/abs/1704.04368.

[17] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative Adversarial Network for Abstractive Text Summarization," p. 2.

[18] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," Multimedia Tools and Applications, Feb. 2018, doi: 10.1007/s11042-018-5749-3.

[19] C. Li, W. Xu, S. Li, and S. Gao, "Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, 2018, pp. 55–60, doi: 10.18653/v1/N18-2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[20] K. Yao, L. Zhang, D. Du, T. Luo, L. Tao, and Y. Wu, "Dual Encoding for Abstractive Text Summarization," IEEE Trans. Cybern., pp. 1–12, 2018, doi: 10.1109/TCYB.2018.2876317.

[21] D. Harman and P. Over, "The Effects of Human Variation in DUC Summarization Evaluation," p. 8.

[22] C. Napoles, M. Gormley, and B. V. Durme, "Annotated Gigaword," p. 6.

[23] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," Expert Systems with Applications, vol. 68, pp. 93–105, Feb. 2017, doi: 10.1016/j.eswa.2016.10.017.

[24] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," p. 10, 2004.

[25] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A Fast and Furious Segmenter for Arabic," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, San Diego, California, 2016, pp. 11–16, doi: 10.18653/v1/N16-3003.

[26] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," arXiv:1508.04025 [cs], Sep. 2015, Accessed: Dec. 16, 2019. [Online]. Available: http://arxiv.org/abs/1508.04025.

[27] D. Suleiman and A. A. Awajan, "Using Part of Speech Tagging for Improving Word2vec Model," in 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, Oct. 2019, pp. 1–7, doi: 10.1109/ICTCS.2019.8923081.

[28] D. Suleiman and A. Awajan, "Bag-of-concept based keyword extraction from Arabic documents," in 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, May 2017, pp. 863–869, doi: 10.1109/ICITECH.2017.8079959.

[29] D. Suleiman, A. Awajan, and N. Al-Madi, "Deep Learning Based Technique for Plagiarism Detection in Arabic Texts," in 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Oct. 2017, pp. 216–222, doi: 10.1109/ICTCS.2017.42.

[30] D. Suleiman and A. Awajan, "Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications," in 2018 International Arab Conference on Information Technology (ACIT), Werdanye, Lebanon, Nov. 2018, pp. 1–7, doi: 10.1109/ACIT.2018.8672674.

[31] D. Suleiman, A. A. Awajan, and W. al Etaiwi, "Arabic Text Keywords Extraction using Word2vec," in 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 2019, pp. 1–7, doi: 10.1109/ICTCS.2019.8923034.

[32] M. K. Saad, "OSAC: Open Source Arabic Corpora," p. 6.

[33] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, 2014, pp. 55–60, doi: 10.3115/v1/P14-5010.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: An example of the generated summary using the proposed model

Source document
أشارت توصيات أصدرتها لجنة مستقلة من الخبراء في مجال الطب إلى أن الأشخاص بين 50 و59 عاماً من المعرضين بصورة أكبر للإصابة بأمراض القلب والسكتة الدماغية عليهم تناول جرعة يومية من الأسبرين المنخفض الجرعة وقالت قوة عمل الخدمات الوقائية الأميركية التي تحظى بمساندة الحكومة إنه علاوة على الحيلولة دون الإصابة بالأزمات القلبية والسكتات الدماغية فإن من يتبعون هذا النظام تقل لديهم فرص الإصابة بسرطان القولون إذا داوموا على جرعات الأسبرين هذه لمدة عشر سنوات على الأقل تأتي هذه التوصيات على نحو أضيق نطاقاً من توصيات سابقة أصدرتها قوة العمل والتي ربطت مقترحاتها بالنوع وبشريحة عمرية مختلفة وتستند تعديلات التوصيات إلى إضافة مخاطر الإصابة بسرطان القولون وإضافة نتائج أربع تجارب إكلينيكية بشأن الأسبرين منذ 2009 وقال دوج اوينز عضو اللجنة من نوصيهم بأن يتناولوا الأسبرين هم الذين يتعرضون لمخاطر متزايدة للإصابة بأمراض القلب والأوعية الدموية ومن هم ليسوا عرضة لمضاعفات النزيف ورفضت الإدارة الأميركية للغذاء والدواء العام الماضي وصف الأسبرين لمنع الإصابة بالأزمات القلبية والسكتة الدماغية من جهة أخرى أوضحت نتائج دراسات أجريت على حيوانات إلى أن تناول مرضى السرطان للأسبرين العادي الزهيد الثمن يمكن أن يقوي فاعلية العقاقير الحديثة الباهظة التكلفة التي تساعد جهاز المناعة لديهم على مكافحة الأورام Recommendations by an independent panel of medical experts indicated that people between 50 and 59 years of age who are at higher risk of heart disease and stroke should take a daily dose of low-dose aspirin and the US-backed Preventive Services Task Force said that in addition to preventing infection With heart attacks and strokes, those who follow this system have a lower chance of developing colon cancer if they keep these aspirin doses for at least ten years. These recommendations come in a narrower range than previous recommendations issued by the labor force that linked a proposal. The type and chip different age and adjustments are based on recommendations in addition to the risk of colon cancer and adding the results of four clinical trials on aspirin since 2009, committee member Doug Owens said, "We recommend that they take aspirin, those who are at increased risk of cardiovascular disease, and who are not exposed to complications of bleeding." The US Food and Drug Administration refused last year to describe aspirin to prevent heart attacks and stroke. On the other hand, the results of studies conducted on animals showed that consuming cancer patients with low-cost regular aspirin can strengthen the effectiveness of expensive modern drugs that help their immune system to fight tumors.
Ground truth summary
الأسبرين يحمي من الجلطات والسرطان Aspirin protects against clots and cancer
Proposed model generated summary
الاسبرين يقلل من السكته الدماغيه Aspirin reduces stroke