Computers, Materials & Continua
DOI: 10.32604/cmc.2022.030090
Article

Tech Science Press

An Intelligent Tree Extractive Text Summarization Deep Learning

Abeer Abdulaziz AlArfaj and Hanan Ahmed Hosni Mahmoud*

Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
*Corresponding Author: Hanan Ahmed Hosni Mahmoud. Email: hahosni@pnu.edu.sa

Received: 18 March 2022; Accepted: 17 May 2022

Abstract: In recent research, deep learning algorithms have presented effective representation learning models for natural languages. The deep learning-based models create better data representation than classical models. They are capable of automated extraction of distributed representation of texts. In this research, we introduce a new tree Extractive text summarization that is characterized by fitting the text structure representation in knowledge base training module, and also addresses memory issues that were not addresses before. The proposed model employs a tree structured mechanism to generate the phrase and text embedding. The proposed architecture mimics the tree configuration of the text-texts and provide better feature representation. It also incorporates an attention mechanism that offers an additional information source to conduct better summary extraction. The novel model addresses text summarization as a classification process, where the model calculates the probabilities of phrase and text-summary association. The model classification is divided into multiple features recognition such as information entropy, significance, redundancy and position. The model was assessed on two datasets, on the Multi-Doc Composition Query (MCQ) and Dual Attention Composition dataset (DAC) dataset. The experimental results prove that our proposed model has better summarization precision *vs.* other models by a considerable margin.

Keywords: Neural network architecture; text structure; abstractive summarization

1 Introduction

Text summarization is an important research topic in language processing. It is an ideal way to attack the information surplus challenge by dropping the size of lengthy text(s) into a less number of phrases or fewer paragraphs. The status of mobile devices, such as smart tablets, marks text summarization as a crucial tool for small screens and less bandwidth capabilities [1–3]. It can also be used as a comprehension exam for computers. To produce an acceptable summary, it is essential for a deep learning method to comprehend the text(s) and condense the significant information from it. These responsibilities are highly problematic for computers when the text size increases. Although search engines are allowed to use advanced retrieval methods, they don't have the capacity to extract data from several sources and to return a brief helpful response. Also, there is a necessity for timely tools to abstract several sources. These alarms have started an interest in computerized text summarization models. Current text summarization techniques depend on refined feature extraction engineering that used statistical features of the text. these systems are complicated, and need engineering models. Also, those systems fail to generate an understandable text summary. End-to-end training models demonstrate better results in other aspects, such as face recognition, machine translation and object recognition. Currently, neural summarization models gained attention; several techniques are proposed and their uses to text corpus were demonstrated [4–7].

There are two models of neural summarization, extractive summarization and abstractive summarization. Extractive models decide and join relevant phrases from a document to generate the summary while keeping its original content. Extractive models are usually utilized for real-world applications [7–10]. A central issue in an Extractive model is to determine the salient phrases that define the key information [11]. While, Extractive summarization models built a semantic model for the original text and produce a summary resembles a human one. The current abstractive models are very weak [12–15].

Still, neural models have some problems while used in text summarization. These models lack the underlying aspect structure of document contents. Therefore, the text summary uses only representation vector space that does not capture multi-aspect content [16]. Another problem is that neural architectures are modifications of recurrent networks such as Gated unit and term memory. These networks can theoretically remember previous selections in the computed state vector. While in reality, this is not the situation. Also, remembering the document semantics is relatively difficult and not essential [17]. A weighted representation vector of prior states will be utilized as an additional input to the step that determine the next state. Thus, the model can attain a state computed in past steps, so the last state will save previous states' information [18].

The key contribution of this research is introducing a summarization neural-based model for extraction of relevant phrases from a text by considering the summarization as a classification process. The model calculates the score for each phrase as a phrase-member by extracting features such as content, significance, redundancy and position. Our proposed model has improved efficiency and accuracy: (i) first it uses a tree text representation; (ii) while constructing the text tree, two self-attention techniques are used at words and phrases. This allows the model to respond strongly to important content.

In this research, two problems are arising: (1) reflecting the text tree to enhance the embedding structure of phrases to learn the text semantic; (2) extracting the most significant phrases from the text to produce a preferred summary [18–22].

The main difference between our research and others: is that our proposed model utilizes a tree structure self-attention technique to produce phrase embedding. The augmented self-attention technique can enhance performance and provides understanding to the high score phrases in the summary phrases selection. To test our model performance comparing to state-of-the-art models, two datasets are utilized, the MCQ news, and the DAC set. Our model outperforms the compared models by a significant margin.

The article is organized as follows. Section 2 describes our summarization model in details. Section 3 presents the simulation results. The related models for comparison is described in Section 4. Discussion and conclusions are presented in Section 5.

2 The Proposed Model

Recurrent convolutional neural variants (RCN), have been utilized in text summarization process. To extract text tokens to be fed to these neural networks as inputs. Phrase embedding are utilized to transform text tokens to space representation vectors. Also, attention techniques [23] deem these extractive models scalable, letting them focus on previous portions of the input while deciding the final output.

Assume a text T of a chain of phrases $(P_1,\ P_2,\ \ldots,\ P_n)$ and words $(w_1,\ w_2,\ \ldots,\ w_m)$. Phrase extraction process is used to generate a text summary from T by choosing a subset of K phrases ($K <$ n). we classify the label of the j^{th} phrase L_j as (0, 1). The phrase labels in the text summary set are defined as $L = 1$.

To detail our work, we will start with a brief description of the self-attention algorithm. Assume a query Q and an input tokens chain $(t = [t_1,\ t_2,\ \ldots,\ t_n])$, where t_i is the embedded i^{th} token representation vector. We define a function $f\,(t_i : Q)$ that computes the attention score between Q and a token t_i as depicted in [23]. Self-attention technique is an attention technique where the query Q branches from the input chain. Consequently, self-attention technique represents the token dependencies in the same chain. The function $f\left(t_i : t_j\right)$ computes the dependency of t_j on token t_i, where the query Q is substituted by the token t_j.

The authors in [24] exhibits that using of self-attention process gives higher text representation rank, for text taxonomy. In our work, we propose a new tree structured self-attention model recurrent neural model based on Extractive summarization technique [23–27]. Our text summarization platform is appropriate to most models of phrase extraction utilizing distributed input.

Our model proposes a tree self-attention method which represents the tree structure of phrases and texts, where phrases constitute a text. In the proposal model, there are double plane attention, one plane uses words and the other plan uses phrases. Fig. 1 depicts the model architecture.

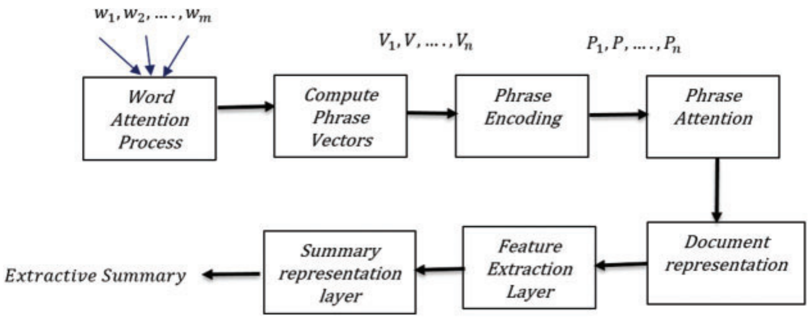

Figure 1: A tree self-attention summarization model: Consists of a word attention process for each phrase that computes phrase representation vectors. After each stage, there is a self-attention layer. The phrase attention layer is the classifier which regulates the phrase affiliation to the summary

2.1 *Word Attention Process*

Given a text document (T) with n phrases $(P_1,\ P_2,\ \ldots,\ P_n)$ and m words and $(w_1,\ w_2,\ \ldots,\ w_m)$, where $P_i\,(1 \leq i \leq n)$ represent the i^{th} phrase, and $V = (v_1,\ v_2,\ \ldots,\ v_m)$ where $v_i\,(1 \leq j \leq m)$ signify a d dimensional representation vector for word embedding for the j^{th} word. In our research, we employ a double pooling RCN (Bi-RC) for word encoding. The Bi-RCN extracts phrases from bi-directions. It encompasses a forward RCN which gets the phrase P_i from v_{i1} to v_{im} and another backward RCN which starts from v_{im} to v_{i1} :

$$\overrightarrow{S_t} = \overrightarrow{RCN}\left(v_t\ ,\ \overrightarrow{S_{t-1}}\right) \tag{1}$$

$$\overleftarrow{S_t} = \overleftarrow{RCN}\ (v_t\ ,\ \overleftarrow{S_{t-1}}) \tag{2}$$

To compute the hidden representative state S_t that recaps the information of a phrase centralized around vi_t, we concatenate $\overrightarrow{S_t}$ and $\overleftarrow{S_t}$ as depicted in the following equation:

$$S_t = (\overrightarrow{S_t} + \overleftarrow{S_t}) \tag{3}$$

where the count of the hidden representative states for each directional RCN is c Let $S_h \in \mathbb{R}^{n\times 2u}$ denotes the whole RCN hidden representative states, as follows:

$$S_h = (S_1,\ S_2,\ \ldots,\ S_n) \tag{4}$$

The attention algorithm pays attention to a word (w_{i}) depending on its impact on the phrase meaning. Our goal is to transform a variable word count phrase into a fixed count using a self-attention algorithm [28–34] that reads the whole S_h as input and produces a representation vector of weights b_h, which are classified by the Softmax classifier using hyperbolic tangent function (tanh) function with learnable parameters.

We use the attention representation vector b_h as a parameter to compute the phrase as a weighted average of the RCN hidden representative states biased by b_h, as depicted in Fig. 2 and in the following equation.

$$h_i = b_{\mathrm{h}}\ S_h \tag{5}$$

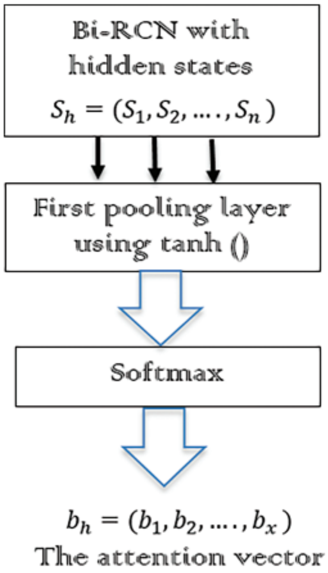

Figure 2: The attention representation

2.2 Phrase Encoding

After obtaining the phrase representation vector P_i, we can develop the text representation in the same technique. A Bi-RCN is utilized to encode the phrases:

$$\overrightarrow{S_{ht}} = \overrightarrow{RCN}\left(h_i\,,\ \overrightarrow{S_{t-1}}\right) \tag{6}$$

$$\overleftarrow{S_{ht}} = \overleftarrow{RCN}\ (h_i\,,\ \overleftarrow{S_{t-1}}) \tag{7}$$

Comparable to the word encoder, the Bi-RCN compute the hidden representative state S_{ht} that recaps the information of adjacent phrases centralized around the phrase P_i, and focus on the ith phrase, as depicted.

$$S_{ht} = (\overrightarrow{S_{ht}} + \overleftarrow{S_{ht}}) \tag{8}$$

where, c denotes the count of the hidden representative states for each directional RCN. Also, N denotes is the number of phrases in the dth text document, and S_d denotes the whole RCN states computed by the following equation (the dimension of S_d is $\mathbb{R}^{n\times 2u}$).

$$S_d = (S_{h1},\ S_{h2},\ \ldots,\ S_{hN}) \tag{9}$$

2.3 Phrase Attention

Each phrase in a text document contributes to the meaning of the text document in a different way and quantify different importance. The self-attention algorithm utilized in this research takes the RCN hidden representative states S_d as input and output a representation vector of weights, b_d, as depicted by the following equation:

$$b_d = Softmax\ (w_{h2} \tanh\left(w_{h1} S_d^T\right) \tag{10}$$

where w_{h1} and w_{h2} are learnable factors. The $Softmax()$ is utilized to normalize the weights to add up to 1.

From the attention representation vector b_d, we obtain the document representation vector as a weighted sum of the RCN hidden representative states weighted by b_d, as depicted in Fig. 2, and also in the following equation.

$$d = b_d\ S_d \tag{11}$$

2.4 Classification

We utilized a classifier that performs a binary classification to predict whether a phrase fits in the output or not. The prediction at the jth phrase is determined by the feature representation. Features such as the richness of the phrase R_j, its prominence in the document $Prom_j$, its originality in the accumulated summary O_j, and its location feature Loc_j. The probability that a phrase should be in the output summary is depicted by the following equation:

While, The Richness of the jth phrase in the text is depicted as follows:

$$R_j = W_{rh_j} \tag{12}$$

The following equation computes the prominence of a phrase in the document $Prom_j$:

$$Prom_j = h_j^T W_h d \tag{13}$$

while the originality of the phrase in the accumulated summary O_j, is computed as follows:

$$O_j = \; h_j^T \, W_r \tanh\left(Summ_j\right) \tag{14}$$

where, $Summ_j$ is the summary state representation and is computed as follows:

$$Summ_j = \sum_{k=1}^{j-1} h_i \; Prob\;(y_i = 1|\; h_i \; Summ_i d) \tag{15}$$

where, y_i is binary (1 or 0) number identifies if the phrase j is contained within in the output summary or not.

The location feature Loc_j. of the phrase in the text document is computed as follows:

$$Loc_j = \; W_{Loc} loc_j \tag{16}$$

where, loc_j is the locational embedding of the phrase computed by the concatenating of the embedding indices of the forward and backward positions of the phrase in the text.

W_h, W_r, and W_{Loc} are learned weights computed from the learning module of the model which represent the relative significance of the features.

From Eqs. (13)–(16), we can deduce the final probability for the phrase with label L_j, as in following equation:

$$Prob\;(y_i = 1|\; h_i, \; Summ_{i,} \;, d) = \boldsymbol{\psi}\left(R_j + Prom_j + O_j + Loc_i + b\right) \tag{17}$$

where $\boldsymbol{\psi}$ is the sigmoid function, and b is the bias weight. The summary state representation, $Summ_{i,}$, in the score lets the model consider the previous decisions by computing the sentence affiliation. At the training stage, the likelihood of the predicted labels is optimized.

3 Experiments

In this following subsections, we are testing the effectiveness of our proposed model starting with describing the used datasets and the experiment setting.

3.1 Datasets

The proposed model was tested on MCQ [35] and DAC datasets [36]. The MCQ dataset was originally constructed for questioning and replying answering process and then was utilized for extractive text summarization process [36–39]. From the MCQ dataset, we utilized 198,240 text documents for training, 23,137 text documents for validation and testing. In the combined MCQ and DAC dataset, we used 296,322 for training, 23,313 for validation testing. The mean number of phrases per document is 34. One of the main contributions of the authors in [40–42] is that they prepared the combined MCQ/DAC dataset for extractive summarization. They provided phrase-level labels for each text document, defining the membership score of the phrases.

The DAC dataset is utilized as not-in-domain test dataset. It has 513 articles fitting in 79 clusters of different topics, and the equivalent 130-word human-made summaries produced for each text (single-text), or the 130-word handatad-made multi-text summaries produced for 63 document clusters. In our model, we employ the single-text summarization process.

3.2 Setting

There are several methodologies for text summarization; for comparison, we select those that are analogous to our model employing the selected datasets as:

- Leading phrases (LP4) [43]: which generates the first four phrases of the text as a summary. This model is considered as our base model on the MCQ and DAC datasets.
- M1: Recurrent CNN (R-CNN) presented in [34], is employed as a base model on the two datasets.
- The extractive algorithm proposed in [3] is employed as a base model on the two datasets.
- On MCQ dataset, we employed a combination of the abstractive model proposed by the authors in [13] and a pointer-Network in [15] as an abstractive base model (M2).
- We also used the topology graph phrase model (M3) [44], which is a graph model as base models for the DAC dataset as they reach better accuracy on DAC.

We started the word embedding process by pre-training on the MCQ dataset. The validation subset was utilized to tune the parameters. The embedding state dimension was fixed to 130 and the hidden state was settled to 230. The concatenation of the Bi-RCN yields a dimension of 460 for word and phrase encoders. Both of them have an attention content vectors of dimension 460. The vocabulary count was bounded to 153 k word. We fixed the maximum phrase length to 53 words and the maximum count of phrases per text to 130. At training, the batch included 64 documents, and CNN [19] was utilized for training. At testing, we ordered the output prediction probabilities for the phrase score membership and then select the phrases with the highest probabilities until we reach the compression final rate.

3.3 Model Evaluation

In this research, the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics (R) [19] are employed for the evaluation of the produced summaries. ROUGE metrics compare n-grams of the summary to be tested and human summarized reference. Rouge is calculated as follows: as in Eq. (20).

$$R_N = \frac{\sum_{S\in summary(R)} \sum_{gram_N \in S} count\,(match\ \ (gram_N))}{\sum_{S\in summary(R)} \sum_{gram_N \in S} count\,(gram_N)} \tag{18}$$

- Note 1: To guarantee that the recall metric will not be biased to the phrase length, we utilize the −173 parameter in ROUGE to trim long text summaries in DAC dataset.
- Note 2: It is observed that all the bases use F1 score as a metric on the whole MCQ dataset as abstractive models learn what time stop producing word for the text summary. To guarantee unbiased comparison, we use the same score.

3.4 Experimental Results

The ROUGE software kit [40] and the software in [41–43] are utilized for testing. This metric is applied with the parameters that are presumed in Note 1 and Note 2. We compared the proposed model with other extractive baselines models using two datasets, MCQ and DAC. The output of the testing phase is compared to the human-produced summaries. The test results, depicted in Tabs. 1 and 2, Figs. 3 and 4, affirm that the proposed model realizes better results. From the results, we can mark the following:

- As depicted in Tab. 1 and Fig. 3, the results indicate that our model outperform other models while using the DAC dataset. This affirms that tree self-attention structure can lead to better phrase and text representations and improves the extracted features that can be utilized to produce high performance.

Table 1: The comparison of the proposed model with baseline models on the DAC dataset

	R1	R2	R3	R4
M1	4.344	4.544	4.448	4.445
M2	4.345	4.545	4.436	4.49
M3	4.494	4.494	4.455	4.479
Our proposed model	6.65	6.47	6.494	6.463

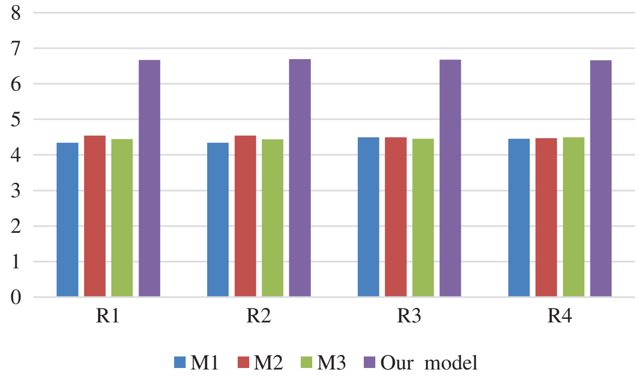

Figure 3: The comparison of the proposed model with baseline models on DAC datasets

- In the case of MCQ dataset, Tab. 2 and Fig. 4 show the high performance of our model *vs.* the baseline models in terms of ROUGE measures.

Table 2: The comparison of the proposed model with baseline models on the MCQ dataset employing full-length rouge (R)

	R1	R2	R3	R4
LP4	3.833	3.953	3.838	3.835
M2	3.335	3.535	3.333	3.39
M3	3.393	3.393	3.355	3.379
Our proposed model	4.45	4.47	4.494	4.443

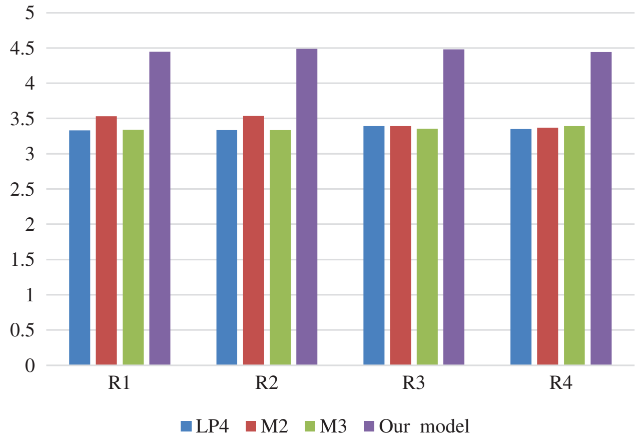

Figure 4: The comparison of the proposed model with baseline models on MCQ datasets

- In many articles, the most significant data is in the beginning. This explains the high ROUGE metric value of the LP-4 baseline in DAC dataset; however, our proposed model has achieved higher performance.
- While ROUGE counts the n-gram state overlap crossing the produced summary and the compared reference, summaries results with better ROUGE values are not essentially the best understandable summary. One issue of automatic summarization techniques is that augmenting for an exact metric value like ROUGE will not assure an enhance the readability of the produced summary [23–27]. This justify the high ROUGE value of the abstractive summarization technique baselines compared in this work.
- Another matter linked to the ROUGE value is that the consistency of ROUGE score will improve by increasing the count of the reference text summaries for each text document. This rigidity of ROUGE deems the Rouge values on datasets with few reference text summaries for each text document to be less compared to the datasets that have several reference text summaries [25–29].
- It should be noted that our model attained better results than the state-of-the-art extractive summarization methods.

Table 3: The comparative study of our model *vs.* similar summarization models in terms of recall, precision and F-measure

Models	Results			Improvement of our proposed model w.r.t. other models		
	Recall	Precision	F-measure	Recall	Precision	F-measure
LP4	0.37	0.35	0.31	+31.5%	+57%	+70.3%
M1	0.35	0.35	0.31	+31.5%	+50%	+70.3%
M2	0.38	0.39	0.51	+51%	+53%	+51.5%
M3	0.37	0.33	0.355	+57.8%	+55.5%	+55.5%
Our proposed model	0.57	0.58	0.597	-	-	-

We conducted a comparative study of our model *vs.* similar summarization models in terms of recall, precision and the harmonic average of precision and recall (F-measure).

3.5 Computational Time

Computational time comparison is founding a good basis to compare summaries construction. We compared our model other extractive text summarization models: LP4, M1, M2 and M3. We compared the time complexity for the models while using MCQ and DAC dataset individually. (Figs. 5 and 6). As illustrated from the figures our proposed model exhibits a benefit in time cost needed to excerpt a phrase in the output summary.

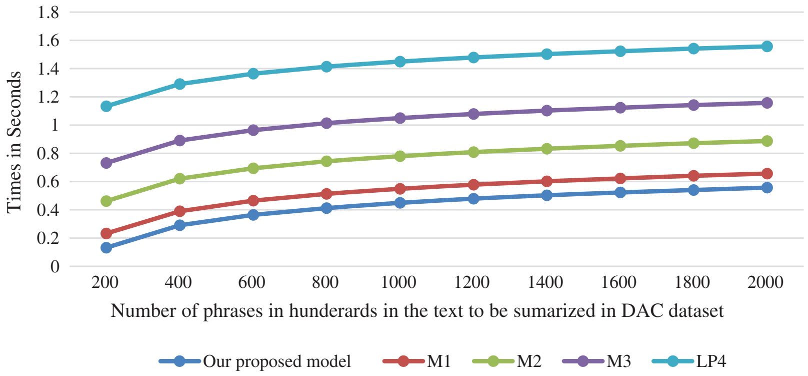

Figure 5: Mean time cost to select a phrase in the output summary in seconds *vs.* count of phrases in hundreds in the text document to be summarized for DAC dataset

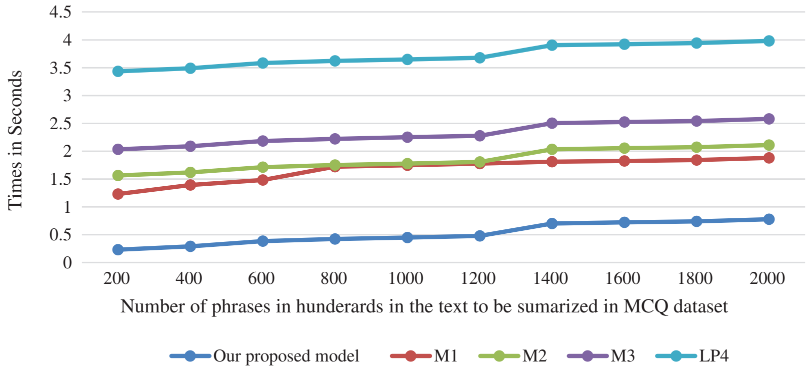

Figure 6: Mean time cost to select a phrase in the output summary in seconds *vs.* count of phrases in hundreds in the text document to be summarized for MCQ dataset

Experiments proved that the computational time cost of selecting N phrases in the output summary has a time complexity in the order of (N Log (N)) where N is the number of phrases in the text documents before summary. Training time is *of time complexity in the order of M* (N Log (N)), where M is the number of all texts used in the summarization training process. We computed the time cost that our model needed in training. to form the final summarization text. as depicted in Fig. 7.

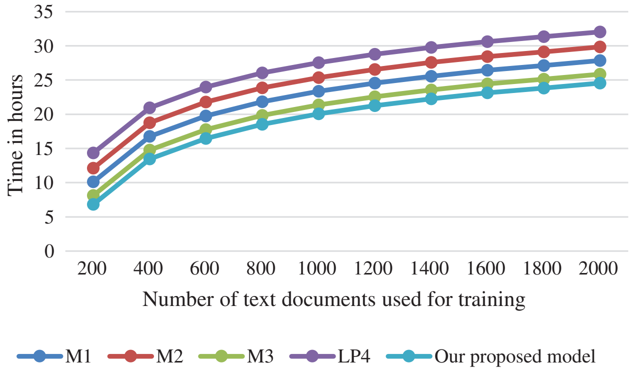

Figure 7: The cost of training time for different models

3.6 Summarization Time Cost

The time cost required to answer a summarization demand is defined as the phrases that the model has to visit. For 2500 requests we counted the phrases selected per request. Fig. 8 illustrates the results for phrases of different number of words. It can be proven that answering a request, our model will navigate a smaller portion of the dataset than other models.

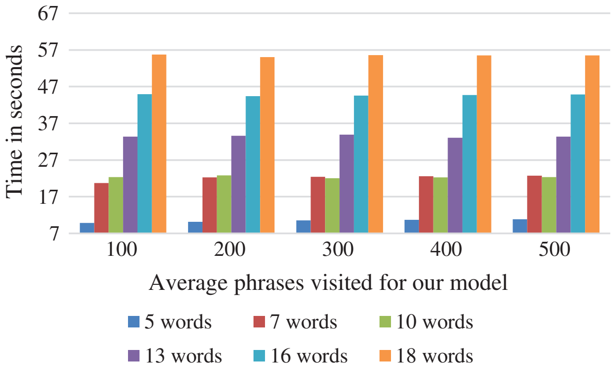

Figure 8: Average response time of our proposed model

4 Conclusions

Our proposed model is using the attention technique to employ phrase embedding. The experiments depict high performance results for our model. Our results emphasize that the self-attention resultant embedding provide better state representation and improves the summarization quality. Our model achieves higher performance than similar models on MCQ and DAC datasets. This work is unlike other models in three aspects. First, it utilizes the tree attention model that reflects better document structure. Second, it utilizes the self-attention module, that generates effective embedding theme. Third, the extracted features are rooted and weighted for the learning phase taking in attention the past classified phrases. We trust that incorporating the support learning with phrase-to-phrase training goal is a motivating path for future research. Another effort has to be focused in suggesting new evaluation metric other than ROUGE score to improve the summarization task particularly for lengthy phrases.

Acknowledgement: We would like to thank for funding our project: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

[1] H. Chieu and L. Teow, "Combining local and non-local information with dual decomposition for named entity recognition from text," in *Information Fusion (FUSION) 2012 15th Int. Conf.*, pp. 231–238, 2012.

[2] H. Silber and K. McCoy, "Efficiently computed lexical chains as an intermediate representation for automatic text summarization. co," *Computational Linguistics*, vol. 28, no. 4, pp. 487–496, 2002.

[3] I. Mani and M. Maybury, "Advances in automatic text summarization," *Computational Speech Language*, vol. 3, no. 1, pp. 126–144, 2019.

[4] X. R. Zhang, J. Zhou, W. Sun, S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.

[5] X. R. Zhang, X. Sun, W. Sun, T. Xu, P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[6] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: A survey," *Artificial Intelligence Review*, vol. 2, no. 1, pp. 1–66, 2016.

[7] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Language-Independent Text Summarization Branches Out: Proc. of the ACL-04 Workshop*, vol. 8, 2004.

[8] K. Sarkar, "Automatic single document text summarization using key concepts in documents," *Journal of Information Processing Systems*, vol. 9, no. 4, pp. 602–620, 2013.

[9] M. Aparicio, P. Figueiredo, F. Raposo, D. Martins, De Matos, R. Ribeiro *et al.,* "Summarization of films and documentaries based on subtitles and scripts," *Pattern Recognition Letters*, vol. 73, pp. 7–12, 2016.

[10] D. Patel, S. Shah and H. Chhinkaniwala, "Fuzzy logic based multi text documents summarization with improved phrase scoring and redundancy removal technique," *Expert Systems*, vol. 134, pp. 167–177, 2019.

[11] R. Elbarougy, G. Behery and A. El Khatib, "Extractive arabic text summarization using modified PageRank algorithm," *Egyptian Informatics Journal*, vol. 21, no. 2, pp. 73–81, 2020.

[12] M. A. Fattah and F. Ren, "Ga MR FFNN PNN and GMM based models for automatic text summarization," *Computational Speech Language*, vol. 3, no. 1, pp. 126–144, 2009.

[13] R. Belkebir and A. Guessoum, "A supervised approach to arabic text summarization using Adaboost," *New Contributions in Information Systems and Technologies*, vol. 1, no. 3, pp. 227–236, 2015.

[14] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms," *Cognitive Computation*, vol. 10, no. 4, pp. 651–669, 2018.

[15] A. Nenkova and K. McKeown, "A survey of text summarization techniques," *Mining Text Data*, Boston, MA, USA: Springer, vol. 2, no. 1, pp. 43–76, 2012.

[16] A. Qaroush, I. A. Farha, W. Ghanem, M. Washaha and E. Maali, "An efficient single text documents Arabic language-independent text summarization using a combination of statistical and semantic features," *Journal King Saud University Computer Informatics*, vol. 2, no. 1, pp. 120–129, 2019.

[17] V. Gupta and G. S. Lehal, "A survey of language-independent text summarization extractive techniques," *Journal of Emergent Technology*, vol. 2, no. 3, pp. 258–268, 2010.

[18] P. Modaresi, P. Gross, S. Sefidrodi, M. Eckhof and S. Conrad, “Benefits of automatic text summarization systems in the news domain: A case of media monitoring and media response analysis,” *Computational Speech Language*, vol. 2, no. 1, pp. 226–2334, 2019.

[19] R. Paulo, K. Margarido, A. Thiago, A. Rachel, M. Aluísio *et al.,* “Automatic summarization for text simplification: Evaluating text understanding by poor readers,” *Cognitive Computation*, vol. 1, no. 4, pp. 51–69, 2019.

[20] D. Andr’e and F. Martins, “A survey on automatic text summarization,” *Computational Speech Language*, vol. 1, no. 2, pp. 126–144, 2019.

[21] C. Brandel and L. Rino, “Pruning UNL texts for summarizing purposes,” *Deartmento de Computacao Universidade Federal de Sao Carlos*, vol. 2, no. 1, pp. 145–154, 2021.

[22] S. Pandian, K. Lakshmana and A. Kalpana, “Unl based document summarization based on levels of users,” *International Journal of Computer Applications*, vol. 6, no. 1, pp. 120–129, 2014.

[23] V. Sornlertlamvanich, T. Potipiti and T. Chroaenporn, “Unl document summarization,” *Electronic and Computer Technology*, vol. 3, no. 1, pp. 45–56, 2019.

[24] S. Mangairkarasi and S. Gunasundari, “Semantic based text summarization using universal networking language,” *International Journal of Applied Information Systems*, vol. 3, no. 3, pp. 167–175, 2012.

[25] W. El-Kassas, C. Salama, A. Rafea and H. Mohamed, “Automatic text summarization: A comprehensive survey,” *Expert Systems*, vol. 5, no. 1, pp. 89–97, 2021.

[26] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez and C. J. Pérez, “Comparison of automatic methods for reducing the weighted-sum fit-front to a single solution applied to multi-text documents text summarization,” *Knowledge Based Systems*, vol. 1, no. 1, pp. 123–136, 2019.

[27] A. Widyassari, S. Rustad, G. Shidik, E. Noersasongko, A. Syukur *et al.,* “Review of automatic language-independent text summarization techniques & methods,” *Journal King Saud University Computer Informatics*, vol. 1, no. 1, pp. 120–128, 2020.

[28] A. Lins, G. Silva, F. Freitas and G. Cavalcanti, “Assessing phrase scoring techniques for extractive text summarization,” *Expert Systems*, vol. 40, no. 14, pp. 5755–5764, 2013.

[29] Y. Meena and D. Gopalani, “Efficient voting-based extractive automatic text summarization using prominent feature set,” *IETE Journal of Intelligent Systems*, vol. 62, no. 5, pp. 581–590, 2016.

[30] C. Jung, R. Datta and A. Segev, “Multi-text documents summarization using evolutionary multi-objective optimization,” in *Proc. Genetic Evol. Comput. Conf. Companion*, pp. 31–32, 2017.

[31] A. Al-Saleh and M. Menai, “Automatic arabic text summarization: A survey,” *Artificial Intelligent Review*, vol. 45, no. 2, pp. 203–234, 2016.

[32] BillSum database: https://www.tensorflow.org/datasets/catalog/billsum.

[33] MLSUM database: https://github.com/huggingface/datasets/tree/master/datasets/mlsum.

[34] Arabic database: https://github.com/mawdoo3.com.

[35] K. Harazin, “Mcq: Multi-text documents arabic text summarization,” *Multi-Text Documents Arabic Text Summarization*, vol. 2, no. 1, pp. 23–38, 2020.

[36] A. Al-Saleh and M. Menai, “Ant colony system for multi-text documents summarization,” in *Proc. 27th Int. Conf. of Computational Linguistics*, pp. 734–744, 2018.

[37] A. Al-Saleh and M. Menai, “Solving multi-text documents summarization as an orienteering problem,” *Algorithms*, vol. 11, no. 7, pp. 96, 2018.

[38] V. Patil, M. Krishnamoorthy, P. Oke and M. Kiruthika, “A statistical approach for text documents summarization,” *Journal of Intelligent Systems*, vol. 4, no. 1, pp. 55–64, 2019.

[39] R. Qumsiyeh and Y. Ng, “Searching web text documents using a summarization approach,” *International Journal of Web Information*, vol. 12, no. 1, pp. 83–101, 2019.

[40] A. Pal and D. Saha, “An approach to automatic language-independent text summarization U sing simplified lesk algorithm and wordnet,” *International Journal of Control Theory and Computer Modeling (IJCTCM)*, vol. 3, no. 4/5, pp. 15–23, 2019.

[41] C. Keong, Z. Hu and L. Beng, “Fusion of simplified entity networks from unstructured text,” *Information Fusion*, vol. 2, no. 3, pp. 1–7, 2020.

[42] H. Morita, T. Sakai and M. Okumura, “Text token Pnowball: A co-occurrence-based approach to multi-text documents summarization for question answering,” *Information Media Technology*, vol. 7, no. 3, pp. 1124–1129, 2019.

[43] D. Radev, H. Jing, M. Stys and D. Tam, “Centroid-based summarization of multiple documents,” *Information Processing and Management*, vol. 40, no. 6, pp. 919–938, 2019.

[44] F. Flores and V. Moreira, “Assessing the impact of stemming accuracy on information retrieval-a multilingual perspective,” *Information Processing & Management*, vol. 3, no. 3, pp. 125–137, 2020.