A Feature-Augmented Deep Learning Model for Extractive Summarization

Bui Thi Mai Anh and Nguyen Thi Thu Trang(✉)

Software Engineering Department, School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
{anhbtm,trangntt}@soict.hust.edu.vn

Abstract. Extractive text summarization can be seen as a classification task in which sentences from the document are labelled with either *in-summary* or *not-in-summary*. The most salient sentences (i.e., with highest ranking score) from the original document will be selected to generate the summary. Recent success of deep learning in the field of Natural Language Processing (NLP) has raised a trending research direction for text summarization task. Many neural models have been proposed in which applying recurrent neural network (rNN) for extractive summarization is also becoming increasingly popular. In this paper, we aim to improve the baseline sequence to sequence model proposed by Nallapati et al. by augmenting more sentence features so that the generated summary can benefit from potential features of the whole document. On one hand, the additional sentence-based features enrich the representation vector resulting from the sentence-level RNN of the baseline model. On the other hand, the relevant information from word-level will also be added to the final vector to increase the accuracy of the classification task. The experiment has been conducted for the DailyMail/CNN dataset to evaluate our proposed method and the state of the art works. The empirical results show that the proposed model with augmented features increases about 0.3-0.4 points of ROUGE-1 and ROUGE-2 in comparison with related works.

Keywords: Extractive summarization · Sequence to sequence model · Recurrent neural networks

1 Introduction

Text summarization is a process of automatically generating summaries from an input document while preserving the overall meaning as well as the coherence of the original. Automatic text summarization can be categorized into two main approaches: (i) *extractive* and (ii) *abstractive* [10]. The extractive-based approaches form summaries by selecting the most salient sentences from the input document and assembling them in a coherent way. In contrast, the abstractive-based approaches aim to generate *human-written* summaries by

N.-S. Vo et al. (Eds.): INISCOM 2021, LNICST 379, pp. 276–286, 2021.
https://doi.org/10.1007/978-3-030-77424-0_23

applying linguistic methods to rephrase sentences which convey the most important information of the original document. Although the abstractive summarization is better than the extractive one, it requires to understand the semantic as well as the structure of the input document, which in turn results into a difficult task for a computer. The extractive summarization, therefore, has attracted much more attention of researchers. An extractive method can be considered as a classification task, which decides whether or not each sentence of the input document belongs to the summary. Traditional methods mainly focus on scoring sentences using graph-based models [7,15], rule-based methods [3,22] to explore the relation between text components (i.e., words or sentences). Some approaches apply evolutionary algorithms to calculate sentence scores on the basis of relevant sentence features such as relation to title document, sentence position, proper noun etc. [1,13,14]. The ranking model is defined as a linear combination of features whose weights are typically determined through experiments.

With the recent emergence of deep neural models, many studies have adopted neural networks for extractive summarization task. Yin and Pei proposed to use *Convolutional Neural Network* (CNN) to learn sentence representation and process the classification of sentences as an optimization problem (in which sentences with high prestige and diverseness would be selected) [24]. This work was however devoted to multi-document summarization. In another work of Cheng et al., a CNN was also applied to encode sentences together with a *Recurrent Neural Network* (RNN) to represent the input document as a sequential of encoded sentences [6]. The classification task was performed by another RNN to take into account previously labeled sentences. Nallapati et al. proposed a similar architecture to encode documents at both word- and sentence-levels [16]. The labelling task is then explicitly enhanced with document features including sentence content, salience, novelty and position.

It is observable that the RNN architecture matched and outperformed the state of the art studies for extractive summarization [6,16,24]. We however argued that the ability of RNNs to take into account previous information of only last few steps to describe the present state might affect the quality of generated summaries. Indeed, it is likely that the relevant information which might be stored at some first words of the underlying sentence cannot be taken into account to represent the state of this sentence. In this case, the probability to be chosen for the summary of the sentence might be reduced. Moreover, the ability to capture the whole context of the document, which could be regarded as a shortcoming of the sequence to sequence architecture, is also required to improve the saliency of the generated summary. In order to address this problem, we propose to integrate several context features so that the summary representation can be augmented with the whole sentence features. More concisely, we adopt two bi-directional RNN models to describe the input document at both word and sentence levels, inspiring from the work of Nallapati et al. [16]. The probability of being selected for the summary of each sentence is then augmented with six proposed word/sentence-based features (see details in Sect. 3). With these features, the meaning of the whole input sentences would contribute to the output result of the RNN, increasing the accuracy of generated summaries.

The rest of this paper is organized as follows. We first introduce closely related works on the topic of analyzing sentences features for improving summarization techniques in Sect. 2. Section 3 describes our proposed approach with some improvements based on six word/sentence features. We finally present the empirical results in Sect. 4 and conclude the paper with remarks on our future direction in Sect. 5.

2 Related Work

This section gives a short overview of related works under the context of applying deep learning in automatic text summarization.

Rush et al. [20] were the first to successfully apply deep learning to text summarization task. They proposed an attentional-based sequence to sequence model to generate abstractive summaries. Inspired by the neural machine translation model [2], novel words of the summary are generated on the basis of a conditional probability on encoded input sentences. Some following works on neural abstractive models of Nallapati et al. [17], See et al. [21], Paulus et al. [19] focused particularly on addressing the out-of-vocabulary problem and on improving the degree of freedom when generating novel words for summaries.

Recent studies have considered also neural networks for the extractive summarization task yielding promising results [4,6,12,16,18,23,24]. Most extractive summarization models are typically processed as a binary-classification in which each sentence of the original document is labeled either *in-summary* or *not-in-summary*. In the same fashion, neural extractive summarization models assign to each sentence a probability of being selected to the output summary. CNNs were first proposed for encoding input sentences with the aim of extracting the most relevant information from the document [24]. RNN models were then adopted for the classification task to take into account previously labeled sentences which are also importance for deciding whether or not the current sentence should be selected for the summary [6]. Wu et al. [23], Narayan et al. [18] then proposed to improve the RNN extractive model with reinforcement learning for the better discrimination among sentences when building summaries. Nallapati et al. have replaced the CNN architecture by two bi-directional RNN models (i.e., backward and forward RNNs) to capture the document at word-level as well as at sentence-level [16]. The chance of being included in the summary of each sentence is calculated based on the hidden state of such models. The obtained result out-performed or was comparable to the state-of-the-art deep learning models.

In this paper, we consider the similar architecture of RNNs integrating with some word and sentence features to give an extra reward to each sentence. Sentences with high rewards are more likely to be selected for the generated summary. Our proposed model will be detailed in the next section.

3 Proposed Approach

3.1 Baseline Model

The baseline model of our work is the SummaRuNNer model proposed by Nallapati et al., the state-of-the-art one for neural extractive summarization [16]. In this model, extractive summaries are generated through a two-layer bi-directional GRU-RNN. The input sentences from the original document are sequentially passed into the model. The first layer, working at the word-level, aims to extract the relationship between words of each input sentence. The second layer, which takes the hidden states of the first layer as inputs, focused on encoding the representation of the whole sentences of the document. The hidden states of the second layer rNN are then used to represent the entire document through a non-linear transformation (i.e., a `tank` function). The classification layer decides whether or not a sentence belongs to the summary through a probability as shown in Eq. 1.

$$\begin{aligned} P(y_j = 1 | h_j, s_j, d) = \sigma(W_c h_j & \texttt{ \#(content)} \\ + h_j^T W_s d & \texttt{ \#(salient)} \\ - h_j^T W_r tanh(s_j) & \texttt{ \#(novelty)} \\ + W_{ap} p_j^a & \texttt{ \#(abs. pos. imp.)} \\ + W_{rp} p_j^r & \texttt{ \#(rel. pos. imp.)} \\ + b), & \texttt{ \#(bias term)} \end{aligned} \tag{1}$$

where the binary variable y_j indicates whether or not the j-th sentence belongs to the summary. h_j denotes the encoding representation of the j-th sentence and s_j dynamically represents the current summary at the j-th sentence position, given by:

$$s_j = \sum_{i=1}^{j-1} h_i P(y_i = 1 | h_i, s_j, d) \tag{2}$$

3.2 Proposed Model

The problem of the RNNs is that they practically can only look back a last few steps. In other words, they cannot store a large number of hidden states of previous steps. Hence, the presentation of the j-th sentence at the j-th hidden state (h_j) cannot represent all necessary information of that sentence.

In this paper, we propose to augment more sentence features so that the summary presentation can be augmented with the whole sentence features, illustrated in Fig. 1. After doing some empirical experiment, we propose to augment six features which can be computed at the word and sentence level. The word-level feature score WF_j is computed from the word sequences of input sentences. The sentence-level features are explored after the second layer of the model, i.e. Sentence-level GRU. These two types features are then combined with SummaRuNNer ones to build up a complex feature layer. The sentence distribution

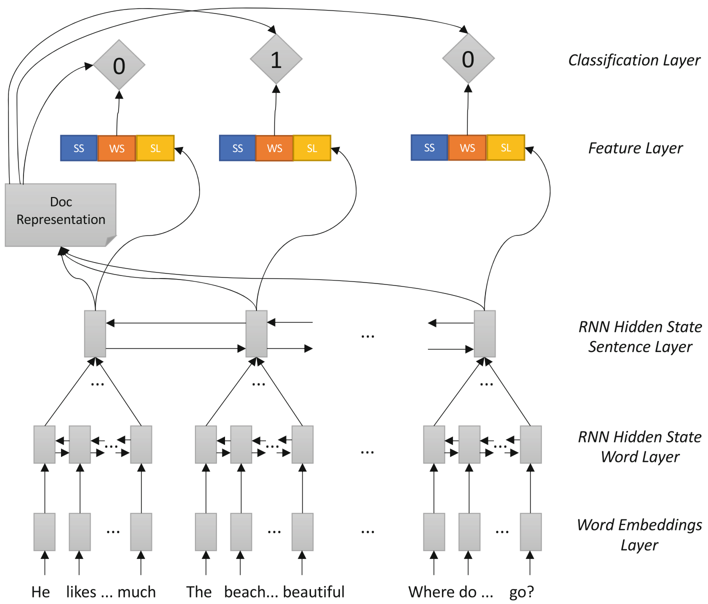

Fig. 1. Proposed extractive summarization model with augmented sentence and word-level features

is modeled and computed based on these features with their weights. This can be viewed as a probability distribution over the input sentences, which represents the output summary.

The logistic layer to determine whether a sentence belongs to the summary (see Eq. 1) now becomes:

$$\begin{aligned}
P(y_j = 1|h_j, s_j, d) = \sigma(W_c h_j \ & \texttt{\#(content)} \\
+h_j^T W_s d \ & \texttt{\#(salient)} \\
-h_j^T W_r tanh(s_j) \ & \texttt{\#(novelty)} \\
+W_{ap} p_j^a \ & \texttt{\#(abs. pos. imp.)} \\
+W_{rp} p_j^r \ & \texttt{\#(rel. pos. imp.)} \\
+WF_j \ & \texttt{\#score of word-} \\
& \texttt{based features} \\
+SF_j \ & \texttt{\#score of sentences-} \\
& \texttt{based features} \\
+b), \ & \texttt{\#(bias term)}
\end{aligned} \tag{3}$$

where SF_j is the augmented feature score at the sentence-level and WF_j is the one at the word-level for the j-th sentence.

The sentence features which can be computed at word-level WF_j is given by individual feature scores and their corresponding weights:

$$WF_j = W_{tfisf} TFISF_j + W_{sl} SL_j + W_{sr} SR_j \tag{4}$$

where $TFISF_j$ is Term Frequency-Inverse Sentence Frequency (TF-ISF) of the sentence, SL_j is the sentence length feature, and SR_j is the stop word ratio of the j-th sentence.

In the same manner, the sentence features which can be computed at word-level SF_j is given by:

$$SF_j = W_{fd}FD_j + W_{pr}PR_j + W_{sf}ST_j \tag{5}$$

where whether FD_j the sentence is the first in the document, FR_j is the sentence score using PageRank algorithm [9], and ST_j is the similarity of the sentence to the topic sentence. These sentence features will be explained in detail in the next sub-sections.

3.3 Word-Based Features

Term Frequency - Inverse Sentence Frequency. When investigating each sentence in a single document, a variant of TF-IDF (Term Frequency – Inverse Document Frequency) is typically proposed, TF-ISF (Inverse Sentence Frequency) [5], in which the word frequencies are investigated at the sentence level. A sentence with higher TF-ISF might contain more meaningful information, therefore could be a good candidate in summary. Given a document D, the TF-ISF score of the j-th sentence can be calculated by summing TF-ISF of all words as indicated in Eq. 6.

$$TFISF_j = \sum_{i=1}^{N}(TF_{ij} * log(\frac{M}{sf_j})) \tag{6}$$

where TF_{ij} is the number of occurrences of the i-th word in the j-th sentence, sf_i (sentence frequency) is the number of sentences in the document D contains the i-th word, M is the number of sentences in document D and N is the number of words in the jth sentence.

Sentence Length. The length of a sentence can be considered as an useful feature. Indeed, recent studies have indicated that short sentences are less likely to appear in the summary [13]. In order to qualify this characteristic, we calculate the length (measured by words) of the j-th sentence in the document D as in Eq. 7.

$$SL_j = \frac{L_j - \mu}{\sigma} \tag{7}$$

where L_j is the number of words of the j-th sentence, μ and σ are the average length and standard deviation of all the sentences in the document respectively.

Stopword Ratio. Stop words are generally the most common words in a natural language. They are typically used to express grammatical relationships among words of a sentence [8], therefore, have little lexical meaning. A sentence with a high rate of stop words should not be chosen as part of summary. We therefore take into account the stop word ratio of a sentence as a feature. It is computed as the number of stop words in the sentence compared to the total number of stop words in the whole document.

$$SR_j = \frac{SW_j}{SW_D} \tag{8}$$

where SW_j is the number of stop words of the sentence j-th, SW_D is the total number of stop words in the document D.

3.4 Sentence-Based Features

First in Document. As indicated in some recent studies, the most relevant information of the document tends to appear at the beginning part of the document [8]. The lead-3 summarization (which extracts the first 3 sentences of the document) even gave good results. Therefore, we propose to use this feature to emphasize the role of the first sentence in the document.

$$FD_j = \begin{cases} 1 \text{ if } h_j \text{ is the first sentence} \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

where h_j is hidden state of the j-th sentence, h_0 is hidden state of the first sentence.

Sentence Score with PageRank. PageRank is an algorithm to rank the importance of website pages in Google search engine results [9]. Therefore, in this paper, we propose to use PageRank algorithm to measure the importance of sentences in the document, given by:

$$PR_j = \frac{1-d}{N} + d * \sum h_i \in C(h_j) \frac{PR_i}{L(h_j)} \tag{10}$$

where:

– h_1, h_2. . . h_j. . . h_N are hidden states of the first, second,..., j-th,..., N sentences in the document D
– $C(h_j)$ is the collection of the hidden states linking to the hidden state h_j, corresponding to the j-th sentence
– $L(h_j)$ is the number of outbound links on the hidden state h_j
– d is the coefficient, which is empirical found as 0.85
– N is the number of sentences of the document D

Similarity to the Topic Sentence. The topic sentence plays an important role in the document as it summaries the main idea of the document. All the sentences belonging to the summary should be semantically related to the topic sentence. In this paper, we consider the similarity score between a sentence in the document and the topic as a sentence-based feature. The similarity of two sentences, S_i and S_j, is calculated using Cosine as following:

$$similarity_{cos}(S_i, S_j) = \frac{\sum_{k=1}^{M}(w_{ik} \times w_{jk})}{\sqrt{\sum_{k=1}^{M}(w_{ik}^2) \times \sum_{k=1}^{M}(w_{jk}^2)}} \tag{11}$$

where M refers to the total number of terms in the document, w_{ik} denotes the weight of the term k in the sentence S_i and w_{jk} denotes the weight of the term k in the sentence S_j. In almost English articles, the topic sentence is usually appeared as the first sentence of the document. We therefore calculate the similarity to the topic sentence of the j-th sentence as $ST_j = similarity_{cos}(S_j, S_0)$.

4 Experimentation

4.1 Dataset

In this work, we employ the widely used corpus, Daily Mail/CNN dataset [11], which contains online news articles (27.2 sentences or 680.0 tokens on average) paired with gold summaries (human-written with 3.8 sentences or 52.3 tokens on average). The corpus has 277,554 training pairs of origin document and gold summary; 13,367 validation pairs and 11,443 test pairs. Besides, in the corpus, each sentence has a label: either "0" indicating not in summary, either "1" indicating in summary, or "2" indicating either in or not in summary. In this work, we consider the sentence with the label "2" is the sentence not in summary.

4.2 Empirical Settings

In order to train our proposed model, we used a single GTX-1080Ti GPU and fixed the batch size as 32. In the prediction phase, the summaries are generated using a probability threshold of 0.5 (i.e., $P(s_j) > 0.5$ meaning that the sentence s_j is chosen for the summary). We trained both the baseline model and our proposal model with 150k vocabulary for about 9,000 iterations (5 epochs), similar to the 5 epochs required by the baseline model [16]. For all experiments, our model has 200-dimensional hidden states and 100-dimensional word embeddings.

4.3 Experimental Results

Our experiments are performed on two measurement metrics: (i) the standard ROUGE metric on the gold summaries and the generated summary, reporting the F1 scores for ROUGE-1, ROUGE-2 and ROUGE-L (which respectively measure the word-overlap, bigram-overlap, and longest common sequence between the reference summary and the summary to be evaluated), and (ii) the absolute metric using sentence labels. We obtain our ROUGE scores using the `pyrouge`[1].

[1] https://pypi.org/project/pyrouge/.

Table 1. Result on DailyMail/CNN corpus using full-length F1 variants of absolute metrics

Models	Dailymail/CNN (11443 documents)		
	F1(%)	Recall (%)	Precision (%)
SummaRuNNer (re-run)	73.1	69.0	77.6
SummaRuNNer + 3 word-level features	73.1	69.0	77.8
SummaRuNNer + 3 sen.-level features	73.1	69.0	77.8
SummaRuNNer 6 features+	**75.2**	**72.4**	**78.3**

Table 2. Result on DailyMail/CNN corpus using full-length F1 variants of Rouge

Models	Dailymail/CNN (11443 documents)		
	R-1(%)	R-2 (%)	R-L (%)
Cheng et al. 2016 [6]	35.4	13.3	**32.6**
SummaRuNNer + (re-run)	39.7	16.6	30.5
SummaRuNNer + 3 word-level features	40.0	16.8	30.0
SummaRuNNer + 3 sen.-based features	40.0	16.8	30.0
SummaRuNNer + 6 features	**40.1**	**16.9**	30.1

Table 3. Result on Daily Mail corpus use limited length (75 bytes and 275 bytes)

Models	Recall at 75 bytes (%)			Recall at 275 bytes (%)		
	R-1	R-2	R-L	R-1	R-2	R-L
Cheng et al. 2016 [6]	22.7	8.5	12.5	**42.2**	17.3	**38.4**
SummaRuNNer + (re-run)	25.3	10.8	13.5	41.7	17.4	33.9
SummaRuNNer + 3 word-level features	25.2	10.8	13.3	41.8	17.5	34.0
SummaRuNNer + 3 sen.-based features	25.2	10.8	13.3	41.8	17.5	34.0
SummaRuNNer + 6 features	**25.5**	10.9	13.5	42.0	17.6	34.2

We re-run the baseline model, and then compare with the results obtained from our proposal models with augmented sentence features.

Table 1 shows the absolute performance based on labels with Precision, Recall and F1 scores of the proposed model comparing with the baseline model SummaRunNer and some other previous models. The sentence was chosen as in summary if the $P > 0.5$ (after some empirical experiments). The experimental results show that the proposed model with augmented features outperformed about 2–3% to the previous ones in all three metrics of Precision, Recall and F1.

Table 2 shows the ROUGE Scores of the proposed models comparing with the baseline model SummaRunNer and some other models. The evaluation was performed by comparing the generated summaries with the associated gold references, in terms of ROUGE. A sentence was chosen in summary is it is in the top-four sentences and $P > 0.6$ (after some empirical experiments). The experimental results show that the proposed model with augmented features increases about 0.3-0.4 points of ROUGE-1 and ROUGE-2 (F1 score).

Table 3 shows the performance of various models on the DailyMail corpus using the limited length recall variants of Rouge with respect to the abstractive ground truth at 75 bytes and 275 bytes.

5 Conclusion

In this paper, we investigate some words- and sentences-based features to improve the quality of an extractive text summarization model. Our work is based on the baseline model SummaRUNNer, proposed by Nallapati et al. [16]. With these features, the meaning of the whole input sentences has been better captured, rather than relying on only the RNNs. The experimental results on Daily Mail/CNN dataset shows that our proposed model, the performance of the summarization in absolute metric as well as the ROUGE one increases about 2–3% or 0.3–0.4 point compared to the baseline one. In the future, we plan to study more on the DNN network to have a better representation of the whole document. Furthermore, we have argued that the Rouge measures are typically built based on the character matching, thus cannot reveal the semantical similarity (i.e., two sentences with similar meaning may be considered as different according to some lexical differences). Improving the Rouge measures is therefore a promising research direction in the future.

Acknowledgement. This research is supported by Hanoi University of Science and Technology under the project entitled *"Intent Classification and Slot Tagging Dataset Construction and Conversational Model Development"*.

References

1. Anh, B.T.M., My, N.T., Trang, N.T.T.: Enhanced genetic algorithm for single document extractive summarization. In: Proceedings of the Tenth International Symposium on Information and Communication Technology, pp. 370–376 (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335–336 (1998)
4. Chatterjee, N., Jain, G., Bajwa, G.S.: Single document extractive text summarization using neural networks and genetic algorithm. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) SAI 2018. AISC, vol. 858, pp. 338–358. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-01174-1_26
5. Chatterjee, N., Mittal, A., Goyal, S.: Single document extractive text summarization using genetic algorithms. In: 2012 Third International Conference on Emerging Applications of Information Technology, pp. 19–23. IEEE (2012)

6. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (vol.1: Long Papers). pp. 484–494 (2016)
7. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
8. Ferreira, R., et al.: Assessing sentence scoring techniques for extractive text summarization. Expert Syst. Appl. **40**(14), 5755–5764 (2013)
9. Gustavsson, P., Jönsson, A.: Text summarization using random indexing and pagerank. In: Proceedings of the third Swedish Language Technology Conference (SLTC-2010), Linköping, Sweden (2010)
10. Hahn, U., Mani, I.: The challenges of automatic summarization. Computer **33**(11), 29–36 (2000)
11. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in neural information processing systems, pp. 1693–1701 (2015)
12. Liu, Y.: Fine-tune bert for extractive summarization. arXiv preprint arXiv:1903.10318 (2019)
13. Meena, Y.K., Gopalani, D.: Evolutionary algorithms for extractive automatic text summarization. Procedia Comput. Sci. **48**, 244–249 (2015)
14. Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., León, E.: Extractive single-document summarization based on genetic operators and guided local search. Expert Syst. Appl. **41**(9), 4158–4169 (2014)
15. Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions, pp. 170–173 (2004)
16. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: AAAI, pp. 3075–3081 (2017)
17. Nallapati, R., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
18. Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636 (2018)
19. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: International Conference on Learning Representations (2018)
20. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
21. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
22. Suanmali, L., Salim, N., Binwahlan, M.S.: Fuzzy logic based method for improving text summarization. arXiv preprint arXiv:0906.4690 (2009)
23. Wu, Y., Hu, B.: Learning to extract coherent summary via deep reinforcement learning. arXiv preprint arXiv:1804.07036 (2018)
24. Yin, W., Pei, Y.: Optimizing sentence modeling and selection for document summarization. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)