A REVIEW OF EXTRACTIVE TEXT SUMMARIZATION USING BERT AND PAGE RANK

[1] Miss. Kumkum Sunilkumar Adwani, [2] Prof. P. P. Shelke

[1]P.G. Scholar, GCOE Amravati Maharashtra India

kumkum29699@gmail.com

[2]Professor, GCOE Amravati Maharashtra India

shelke.prajakta@gcoea.ac.in

Abstract

In a range of natural language processing (NLP) tasks, BERT, a pre-trained Transformer model, has already established itself as one of the most popular models. Using the BERT model, it has been adjusted for extractive summarization. In contrast to other NLP tasks, extractive summarization significantly relies on document-level information about sentence location. As a result, our strategy makes advantage of BERT, which has thoroughly researched the crucial A Page Rank method is employed, with an initial score for each node and edges between sentences being the cosine similarity between the phrases, to create a final summary that has sentences with more information and that are well related with one another Preprocessing, features extraction, and graph creation are the first three steps in the text summarizing process. The final step is applying the Page Rank algorithm and summary extraction. The extracted summary is reliant on compression ratio, taking into account reducing redundancy based on sentence overlapping, and the PageRank method uses a variable number of iterations to obtain the number that produces the best summary results. The suggested method performs better than previous extractive text summarizing strategies.

Keywords:

Text summarization, Extractive text summarization, Natural language processing, cosine similarity, BERT embedding.

1.Introduction

The desire for automatic text summarization to highlight the most crucial information from the page increased due to the enormous volume of data that is generated every day on the internet since its creation two decades ago. A excellent text summary tool spares the stoner the hassle and time of reading the entire document in order to obtain the requested information (1) Text summary is the technique of minimising the amount of material in order to extract the most crucial passage from the original text and provide it to drug users. The process of summarizing is carried out automatically using automatic text synthesis. The English language was the subject of several text summary studies because it has a straightforward alphabet and structure, in contrast to the English language's complicated morphology and organisation. There are more than 350 million English speakers worldwide, so there is a huge need for English text summary. Depending on the comparison factor, there are many ordering that can be utilised for text summary. Text summarising can be separated into single document and multiple document summarization depending on the quantity of documents. Text summarising can be separated into single document and multiple document summarization depending on the quantity of documents. On the other hand, text summarization can be divided into two categories depending on the type of recaptured rulings: abstractive summarization, which relies on regaining the meaning of the rulings without clinging to judgement structure, and extractive summary, which includes the named rulings from the original text without any revision on the judgement structure. Additionally, it comes in general and query- based summary varieties. The general summary provides decisions without taking into account any queries or connections to the title. Due to the relationship between the summary and the question that was asked or the relationship between the judgement and the document title, the query- grounded summary is returned. English text summarization continues to perform poorly, and there aren't enough studies conducted in this NLP function (3). Additionally, these methods exclude the value of using characteristics (4) in their workshop; as a result, it's critical to create a fresh strategy based on noun data. Features have a direct impact on the judgment's importance (5). The decision is given extra weight by additional features. BERT is used to analyse the English judgement and highlight the best aspects of each judgement in the document in accordance with the complexity of the English language (6). This study examines a novel method for rewarding English text summarization that combines graph proposition and features count in judgments. This approach starts by applying the following normalisation, tokenization, stemming, stop words junking, and morphological analyses are examples of preprocessing techniques. also removing redundancy from the summary and extracting the required features, as well as building the graph and using the PageRank method.

2. Related Work

[1] Mayank Patel and Hritvik Gupta, "Study of Extractive Text Summarizer Using The Elmo Embedding", Fourth International Conference on I-SMAC IEEE Xplore, 2020. This study focuses on extracting useful data from extractive text summarization using Elmo embedding.
[2] English text summary using statistical and semantic analysis. Alami N, El Adlouni Y, En-nahnahi N, Meknassi M. Submitted to: International Conference on Information and Communication Systems. 35–50. Cham: Springer; 2017. A hybrid strategy to produce an

extracted summary of English papers is presented in this research. The method is based on a weighted, undirected, two-dimensional graph. [3] Impact of stemming on English text summarization, Alami N, Meknassi M, Ouatik SA, and Ennahnahi N. In the field of information technology (CiSt). Pp. 338–43 in: 2016 4th IEEE International Colloquium. The purpose of this essay is to assess how three different English stemmers—Khoja, Larekey, and Alkhalil's stemmer—affect the effectiveness of text summary for the English language. [4] Al-Taani AT and RZ Al-Abdallah "Using the Firefly Algorithm to Summarize English Text." IEEE 2019:61-5 in 2019 Amity International Conference on Artificial Intelligence (AICAI). utilises the Firefly algorithm to extract summaries from individual English documents. The proposed strategy is contrasted with two evolutionary strategies that employ harmony search and genetic algorithms. [5] Multilingual Text Summarization based on LDA and Modified PageRank. Malallah S, Ali ZH. 2019;9 (3):139–60. Iraqi Journal of Information Technology. For text summary in this paper, the LDA and Page Rank are used. [6] English text summarization based on graph theory. Alami N, Meknassi M, Ouatik SA, and Ennahnahi N. 12th International Conference on Computer Systems and Applications, 2015 IEEE/ACS (AICCSA). IEEE; 2015. p. 1-8 In order to determine the relative relevance of each sentence in a document, we present a novel method for summarising English text that is based on graph theory and semantic similarity between sentences. [7] Automatic Text Summarization Using TextRank, Tanwi1, Satanik Ghosh1, Viplav Kumar1, Yashika SJain1, Mr. Avinash B, International Research Journal of Engineering and Technology (IRJET), Vol.6, Issue - 04, p-ISSN: 2395-0072, April 2019. We introduce TextRank, a graph-based text processing ranking model, and demonstrate how this model may be successfully applied in natural language applications in this study. [8] Natural Language Processing: State of the Art, Current Trends and Challenges. D. Khurana, A. Koli, K. Khatter, & S. Singh. ArXiv, vol. [abs/1708.05148], published in 2017. The paper divides its discussion into four phases, beginning with a discussion of the various levels of NLP and Natural Language Generation (NLG) components. The history and evolution of NLP, the state of the art, a presentation of the various NLP applications, and current trends and challenges are then covered. [9] “Using ELMo to Improve a Text Summarization System. Fabio Tamburini FICLIT and Claudio Mastronardo DISI both attend the University of Bologna in Italy. CLIC-it conference in Bologna, 2019. Using deep contextualised word embeddings that have been trained beforehand in a pointer-generator network-based sequence-to-sequence architecture, we take use of current developments in transfer learning for Natural Language Processing (NLP). [10]. Matthew E. Peters, Luke Zettleomyer, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, NAACL Word representation with deep contextualization. Association for Computer-Assisted Language Input, Volume 1, 2018 We provide a brand-new kind of deep contextualised word representation that takes into account both the complexity of word use (such as syntax and semantics) and the variation in word use across linguistic settings.

3. METHODOLOGY

A colourful algorithm had been utilised in the past for extracting text summarization automatically. As mentioned in the previous section, the text document is evaluated based on the amount of text and words it includes. Prior to working on extractive text summarization, the experimenters focused on a straightforward instructive point of the text frequency of terms in general that they contain or key expressions that characterised the judgment's relevance. Through summaries that the model generates, this trial hopes to assist drug users in recovering their valuable information and effectively reading the documents. The summary must convey the document's relevant significance. The raw texts are converted to vectors using the BERT embedding in our suggested approach. The bidirectional encoder representation from mills, which captures the environment-dependent parts of the word meaning and aspects of the syntax of the words, is utilised in the BERT embedding model to increase the accuracy of the data. In summary, BERT used to depict the memorial by combining all of these information into an one vector. The cosine similarity function is also used to score the BERT Bedded vector achieved. Each decision will be graded in comparison to other decisions. Once the decisions' scores have been received, the rulings are ordered by being arranged in decreasing order. The top five rulings with the highest scores will then be selected and combined to create the summary.

This approach breaks down the entire process into six steps:

Preprocessing of text

2) BERT feature embedding

3)Sentence rating based on embedded vector cosine similarity

4) Sort the score and initialize the ranks using the Page Rank algorithm.

5) Combining the first few sentences to form a summary.

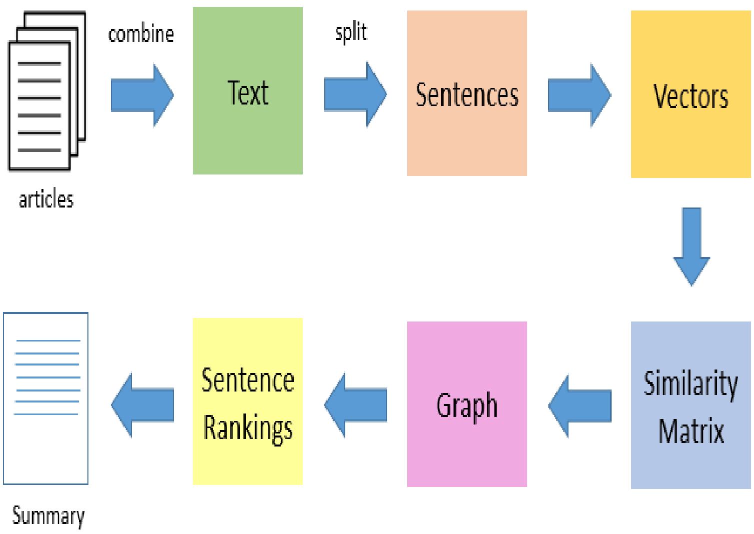

Fig.1. Architecture Of Proposed Approach

A. Text Preprocessing (16) is one of the key NLP jobs for improving and accurately predicting results. The algorithm benefits from directly learning weights. In our model, the document that needs to be epitomized has n rulings, and Elmo Embedding uses this list to create the vectors. Resolving the text document into a list of decisions is the initial stage of preprocessing. There are several ways to break rules into rulings, such as by utilizing NLTK judgement tokenization, however our technique gains additional delicacy when the rulings are resolved from punctuation after the full stop. The next step after dividing is to process the document using NLTK. Each judgement from the list of rulings is processed in one of four ways. Eliminate punctuation. Take each decision from the list, change every word to lower case, and remove all punctuation. One of the main responsibilities is getting rid of punctuation because it makes the model more directly vectorize the input. Punctuation junking entails getting rid of symbols like "?" and "/" A tokenizing of the verdict The most frequent phrase used in natural language processing is "tokenization," which simply refers to grouping the rules into a list of words for additional preprocessing. Thus, each document's judgement is tokenized individually for the purposes of eliminating stop words and lemmatization. This also applies to future methods. Stop using trash words. After tokenizing the judgement, another duty is to eliminate the stop words, such as "the," "if," and so on. Stopping word junking enables the model to record the crucial phrases and words. These stop words are also a major factor in the creation of vague summaries. Lemmatizing This task entails the procedure of combining a word's inflected forms into a single unit so they may be anatomized. It seeks to restore the base while simply removing the inflectional ending. finally merging the remembrances to create the verdict. Lemmatizing is used to shorten the length of judgement and for better data garbling.

3.1 BERT Embedding

BERT or Bidirectional Encoder Representations from Mills it can be salutary to gain contextual embeddings. It generates 768 size vectors for each sub word.

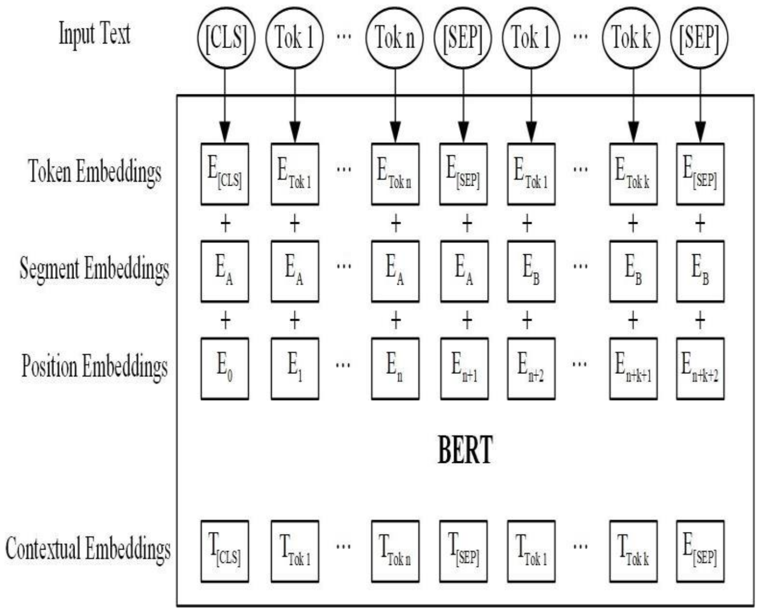

Figure 2. Architecture of the original BERT model. Token [CLS] is appended to the beginning of the sequence, and token [SEP] is inserted after each sentence as an indicator of sentence boundaries.

3.2 **Cosine similarity**

The measure of similarity known as cosine similarity can be used to compare word, legal precedent, or document vectors. Let A and B be two vectors as an example and comparison. -Embedding BERT.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

3.3 Extractive Summarization Models

Fig. 1 displays the BERT's original armature. Preprocessing of the input text sequence includes fitting special commemoratives (CLS) and (SEP). A token (CLS) that summarises the information for the entire series is connected to the morning of the sequence. After every judgement, (SEP) is adapted to show the judgement borders. The changed text is also shown as a series of commemoratives, where each commemorative is a superposition of three different embedding types: token embeddings, member embeddings, position embeddings, and videlicet. These three embeddings independently render the meaning of the token, the boundary between the judgement and the brace, and the position of the token inside the sequence. To obtain an affair vector for each commemoration with contextual information, the additional embedding vector is sent to a multi-layer bidirectional motor. Each commemoration in the sequence is made up of a superposition of embeddings for the member, position, and commemorative. For each commemorative, BERT will generate an affair vector containing contextual data.

3.4 BERT Model

By adding a (CLS) commemorative before each judgement, BERTSUM (18–19) further changed the input sequence and embeddings, drawing inspiration from the sequence preprocessing style in BERT. As a result, let the model to generate many judgments and try to extract judgement characteristics using the (CLS) symbols. Figure 2 depicts the BERTSUM model's structure. In order to improve the judgments, many Transformer layers have been layered on top of the BERT labours after carrying the judgement vectors from BERT. judgement Embeddings in Position The self-attention layer of a Transformer causes similar words to have the same output representation, in contrast to RNN-based sequence models. In order to recover location information, positional embeddings must be added. Two variations of positional embeddings will be covered in this subsection, along with how they are used in extractive summarization models. spatial embeddings in a sine wave. Utilizing sine and cosine functions, sineoidal positional embeddings produce relative position information.

$$\begin{cases} PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{cases}, \qquad \textbf{(2)}$$

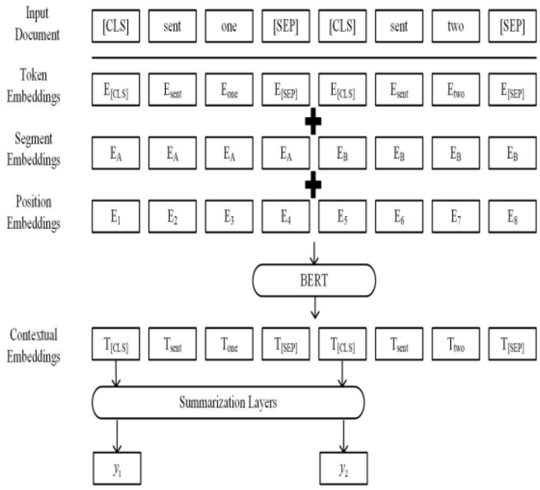

Figure.3. Structure of BERTSUM model. Token [CLS] is appended to the beginning of every sentence.

The (CLS) representations can be used to obtain features of decisions. For calculating the double bracket entropy against the gold marker, rulings scores are forecasted. where pos is the position, model dmodel is the model affair dimension, and I is the number of sinusoids, each of which corresponds to a positional garbling dimension. employing eq. (2), it would enable the model to understand the placements of the commemoratives in relation to one another. Each sinusoid represents a dimension of positional garbling. employing eq. (2), it would enable the model to understand the placements of the commemoratives in relation to one another. positional embeddings were learned. Bed the absolute position indicator with learnable parameters is yet another garbling system of location information. Training is done on the data to obtain the position information of each commemorative given an aimlessly initialized vector of the commemoratives at each position. The model can determine which part of the input sequence is currently being reused thanks to learned positional embeddings, but they also place a limit on how long the input sequence can be. The BERT model used the best positional embedding technique; it learned a vector representation for each position to take into account the successional structure of the input sequences. It was created to allow the reuse of input sequences longer than 512 commemoratives. Accordingly, the Position Embeddings subcaste is a lookup table with a size of (512, model) and a first row that contains the vector representation of every word in the first position, an alternate row that contains the vector representation of every word in the alternative position, etc. Which implies that jobs involving lengthy text sequences, such as documents, may not be appropriate for the basic BERT paradigm. Unless, of course, a structural redesign has begun. Consequently, the BERT model cannot provide documental judgement representations on its own. A BERT model version called BERTSUM makes use of redundant commemorative to obtain each judgement representation of a document in order to enable BERT on the extractive summary task. Additionally, it stacks several Motor layers to obtain a representation of document position judgement. For the documental judgement position data, the authors selected sinusoidal position embeddings. The sinusoidal embedding system merely takes into account the relative placements of rudiments, as has been discussed. The judgement indicator is still the most useful position information in extractive summarization governance. People frequently use the first or last judgement of a document as a summary, for example. Therefore, improving automatic criteria like ROUGE is still possible.

Page Rank formula A extremely intricate network exists on the internet. A graph can be used to illustrate the relationship between the pages. Any vertex's in-degree (out-degree), which is the sum of all inbound and outbound linkages to it, determines its significance. An indication of a page's importance or quality is its inbound links. This concept is used by the PageRank algorithm to rate the pages that show up in search results. Not all inbound links from pages are treated equally by PageRank; some links are given more weight based on how important the page that connects to them is.

3.5 Page Rank algorithm

Input: Weighted Graph G.
Output: Scored Graph.
1 Configure N = Number of Nodes in G.
2 Current_Rank Double[N]
3 Temp_Rank Double[N]
4 Foreach n=1 to N
5 Current_Rank[n] = 1/N
6 For i = 1: Number_of_Iterations
7 Foreach nd: G.Nodes
8 Temp_Rank[nd.index] = Calc_Page_Rank(nd)
9 Current_Rank = Temp_Rank

In Formula (1), where d is the damping factor with a value of 0.85 and N is the total number of bumps, the PageRank algorithm of a Web runner u is denoted as PR(u). The PageRank algorithm is demonstrated with Algorithm 1, which begins by initializing each knot's rank by 1/N, where N is the number of bumps in the graph. Additionally, the algorithm iterates and determines new species of bumps using Formula (1). The species are streamlined after computing the new species for each bump. In accordance with the number of duplicates, this operation is repeated. Since the order of the judgement is directly related to the order of the linked rulings, which varies every time the algorithm is applied, the replication is then utilised to modernise the rank for each judgement in order to obtain the fashionable and most stable rank.

PageRank algorithm changes Arabic nouns have a special value, hence the more nouns a judgement contains, the more significant it is (4). In order to reward Arabic summaries, this investigation modifies the original PageRank algorithm in a novel way that takes into account the quantity of nouns in each judgement. With the following differences, a Modified PageRank differs from the PageRank algorithm in many ways. (1) the document's judgements are substituted for the runners (2) the cosine similarity-based weighting of the edges between bumps, which was not included in the original PageRank.

3) Unlike the original PageRank, which assigns the initial rank inversely to all bumps and equals 1/N, where N is the number of judgments in the document, the original rank of each judgement is the number of nouns in this judgement.

(4) PR (vi) is adjusted in accordance with Formula (2). The new rank of knot is calculated using Formula (2). (5), The totality is eventually divided by the number of the remaining rulings in the document N- 1, which is the number of rulings in the document D with the current judgement banned, to get the new rank of judgement, where PR (vi) is the current rank of judgement (6) and E(g and vi) is the weight of edge connect rulings(g) and(vi), which is also the cosine similarity between these two rulings.

4. CONCLUSION

The (CLS) representations can be used to obtain features of decisions. For calculating the double bracket entropy against the gold marker, rulings scores are forecasted. where pos is the position, model dmodel is the model affair dimension, and I is the number of sinusoids, each of which corresponds to a positional garbling dimension. Each sinusoid represents a dimension of positional garbling. The investigation describes how to use a BERT Embedding for Automatic Extractive Text Summarizer. Through this trial, it is clear that BERT embedding can be used to convert a text file into a vector that has all the information a word needs (contextually dependent elements, grammar, etc.) for an automatic extractive text summarizer. To score and rank the text document's rulings, these vectors are also sent to the cosine similarity algorithm. A good summary can also be produced by selecting the top 5 decisions with the highest score. As can be observed from the recall, which demonstrates the importance of the summary model and how quickly and reliably BERT Embedding can collect the most pertinent information or terms from the text that are contained in the summary. Additionally, the F-measure is visible. When compared to Elmo embedding summarization approach, which uses the BI-LSTM for data garbling, using the BERT Embedding has improved average scores for both longer and shorter articles. In other words, BERT Embedding can be utilized in extractive text summarizers for the purposes of text ranking and summary generation in addition to abstractive text summarizers for summary generation. BERT Embedding has been shown to be an accessible system that can be utilized to create a summary using an extractive system, but there is still room for improvement in terms of the extractive summarizer's refinement and speed of data reuse. Additionally, this summarizer can be improved by processing numerous papers at once and providing a generalized summary. The other improvement that can be made is to use the text's title as its summary. Since the summary's title describes the material in terms similar to what the text is about, using the title will make it more delicate to sum up the text.

REFERENCES:

[1] Khurana, D., Koli, A., Khatter, K., & Singh, S. Natural Language Processing: State of The Art, Current Trends and Challenges. Published ArXiv, vol .[abs/1708.05148] , 2017.

[2] Madhavi K. Ganapathiraju, “Overview of summarization methods”, 11- 742: Self-paced lab in Information Retrievavol,vol 117 November 26, 2002.

[3] L. Dinu and R. Ionescu, "A Rank-Based Approach of Cosine Similarity with Applications in Automatic Classification," in 2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012), Timisoara, 2012 pp. 260-264.

[4] Kulkarni, A. R., & Apte, S. S. (2013). A domain-specific automatic text summarization using Fuzzy , Vol 4, Issue 4, July-August (2013), pp. 449-461

[5] Tanwi1, Satanik Ghosh1, Viplav Kumar1, Yashika SJain1, Mr. Avinash B."Automatic Text Summarization using TextRank" International Research Journal of Engineering and Technology(IRJET), Vol.6, Issue - 04, p-ISSN: 2395-0072 ,Apr 2019
[6] Mathew E peters, mark Neumann, Mohit Iyyer, matt gardner, Christopher clark , Kenton lee, luke Zettleomyer NAACL " Deep Contextualized word representation".Assoiciation of computer linguistics Vol 1, 2018
[7] Vishal gupta, "A survey of text summarization extractive techniques",Journel of emerging technologies in web intellegence, vol 2, No, 3 august 2010
[8] Enhancing a Text Summarization System with ELMo. Claudio Mastronardo DISI - University of Bologna, Italy,Fabio Tamburini FICLIT - University of Bologna, Italy. Bologna conference CLIC-it, 2019
[9] Jing, Li-Ping & Huang, Hou-Kuan & Shi, Hong-Bo. (2002). Improved feature selection approach TFIDF in text mining. 2. 944 - 946 vol.2.
[10] Klaus Zechner, "A Literature Survey on Information Extraction and Text Summarization", Computational Linguistics Program, Carnegie Mellon University, April 14, 1997
[11] Khosrow Kaikhah "Text Summarization using Neural Networks", Department of Faculty PublicationsComputer Science, Texas State University, eCommons,2004. [13] Rasim M. Alguliev and Ramiz M. Aliguliyev, "Effective Summ pp. 960-963 ,2004
[12] Udo Hahn and Martin Romacker, "The SYNDIKATE text Knowledge base generator", Proceedings of the first international conference on Human language technology research, Association for Computational Linguistics, ACM, Morristown, NJ, USA, 2001.
[13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality,in: Advances in neural information processing systems, pp. 3111–3119
[14] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proc. EMNLP Conference in Lisbon, 2015.
[15] J. Gu, Z. Lu, H. Li, V.O.K. Li, Incorporating copying mechanism in sequence-to-sequence learning, in: Proc. ACL Conference of the 54th Annual Meeting, 2016.
[16] A. See, P.J. Liu, C.D. Manning, Get to the point: summarization with pointer generator networks, in: Proc. ACL Conference of the 55th Annual Meeting, 2017.
[17] R. Paulus, C. Xiong, R. Socher, A deep reinforced model for abstractive summarization. , 2017.
[18] S. Narayan, S.B. Cohen, M. Lapata, don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, in: Proc. EMNLP Conference in Brussels, 2018.
[19] H.P. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development (1958) 159-165.
[20] P.E. Baxendale, Machine-made index for technical literature--an experiment, IBM Journal of Research and Development (1958) 354-361.