International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Text Summarization Using Natural Language Processing

Kanithi Purna Chandu

Student, Department of Information Technology, GMR Institute of Technology

ABSTRACT:

Most things in this world have become automated because of technological improvements. The idea of text summarization gained popularity since manually summarizing texts has become complex and time-consuming. Therefore, text summarization's primary goal is to solve the problems associated with manually summarizing text documents or other information from diverse sources. Text summarization means taking the main concepts from the context or text and giving a quick explanation of the context. Text Summarization is categorized into two methods Extractive and Abstractive Summarization. The extractive summarizing method is mainly focused on selecting which paragraphs and key sentences will produce the original materials in their standard form. An abstractive summary is used to comprehend the key ideas in a given document and then to articulate those ideas in understandable, ordinary language. The emphasis of this study is on extractive-based summarization applying TFIDF (Term Frequency-Inverse Document Frequency) and K-Means Clustering.

Keywords: Text summarization, Natural Language Processing, Extractive summarization, Abstractive summarization.

INTRODUCTION

Natural Language Processing (NLP) is a field of computer science, and is a subset of AI. Generally, it concerns with the interactions between machines and human languages. Natural language processing is a process of developing a system that can process and produce language as effectively as humans can produce.

Text summarization a process of reducing the size of the original document while preserving its information content and its summary is less than half of the main text. There are two approaches one is extractive and other is abstractive Extractive summarization generates summaries by extracting important sentences or phrases from the original text, whereas abstractive summary, analyses the important concepts in a given document and then expresses those concepts in understandable natural language. Usually, Term frequency and length of the terms are taken into consideration for minimizing the text. To achieve this text summarization, we use various models like Learning Free Integer Programming Summarizer (LFIP-SUM), attention-based bidirectional LSTM model, Seq2Seq + Attention, enhanced semantic network (ESN).

LITERATURE SURVEY

In paper [1] J. Jiang et al., proposes four novel ATS models with a Sequence-to-Sequence (Seq2Seq) structure, utilizing an attention-based bidirectional Long Short-Term Memory (LSTM), with added enhancements for increasing the correlation between the generated text summary and the source text, and solving the problem of out-of-vocabulary (OOV) words, suppressing the repeated words, and preventing the spread of cumulative errors in generated text summaries. Experiments conducted on two public datasets confirmed that the proposed ATS models achieve indeed better performance than the baselines and some of the state-of-the-art models considered.

In paper [2] H. Gupta and M. Patel, has shown an experiment that contrasts the extractive text summarizer for text summaries. On the other hand, topic modelling is an NLP task that extracts the relevant topic from the textual document. One of the method is Latent semantic Analysis (LSA) using truncated SVD which extracts all the relevant topics from the text. This paper has demonstrated that the experiment in which the proposed research work will be summarizing the long textual document using LSA topic modelling along with TFIDF keyword extractor for each sentence in a text document and also using BERT encoder model for encoding the sentences from textual document in order to retrieve the positional embedding of topics word vectors.

In paper [3] K. Shetty and J. S. Kallimani, proposed an approach for generating an appropriate summary, the raw text is first pre-processed which involves - removing non-ASCII characters and stop-words, tokenizing and stemming. Appropriate features are extracted from the data, tf-idf values for each word are computed and the entire pre-processed data is then transformed into a tf-idf matrix. Every sentence of the document will be represented as a vector in the dimensional space of the document's vocabulary. To obtain a concise summary, sentences are appropriately clustered based on the degree of separation of vectors in the Euclidean place. Association of the sentences to cluster using K-means method is totally based on cosine similarity. The count of the clusters is to be formed is predefined. As the number of clusters increases the accuracy of the summary increase. From each of the clusters the sentences which are informative are extracted to form the final summary. Using recall and precision measures, the effectiveness of the summary is verified.

In paper [4] M. Y. Saeed, M. Awais, R. Talib and M. Younas, proposed approach analyses the mesh of multiple unstructured documents and generates a linked set of multiple weighted nodes by applying multistage Clustering. They have generated adjacency graphs to link the clusters of various collections of documents. This approach comprises of ten steps: pre-processing, making multiple corpuses, first stage clustering, creating sub-corpuses, interlinking sub-corpuses, creating page rank keyword dictionary of each sub-corpus, second stage clustering, path creation among clusters of sub-corpuses, text processing by forward and backward propagation is for results generation. The result for this technique is consisting of interlinked sub-corpuses through clusters.

In paper [5] M. Jang and P. Kang, proposes a model called Learning Free Integer Programming Summarizer (LFIP-SUM), which is an unsupervised extractive summarization model. The advantage of our approach is that parameter training is unnecessary because the model does not require any labelled training data. To achieve this, we formulate an integer programming problem based on pre-trained sentence embedding vectors. they used principal component analysis to automatically determine the number of sentences to be extracted and to evaluate the importance of each sentence.

METHODOLOGY

Bi-LSTM + Attention model:

The model consists of five layers [1]: an input layer used to input data into the model, an embedding layer used to map each word in a low-latitude space, a Bi-LSTM layer that uses a bidirectional LSTM for advanced feature extraction and weight vector generation, an attention layer that is able to generate sentence-level features by combining a weight vector with lexical features, and an output layer. Basically, the 'Bi-LSTM + Attention' model adds an attention mechanism on top of the Bi-LSTM layer, for the purposes of applying an attention weighting. The added attention mechanism can differentiate the weight of each word and thus enable the whole sequence to get the key information more easily.

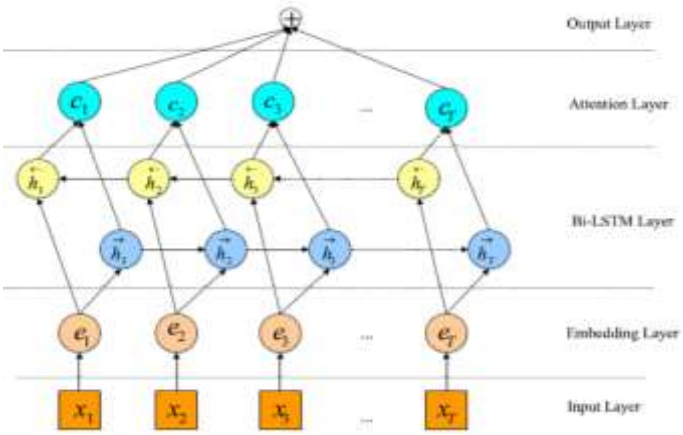

Fig : Bi-LISTM

LSA (latent semantic analysis):

LSA topic modelling using Truncated SVD which is used to extract the topic by generating the matrix that represent conceptually related sentences or documents and corresponding words. TF-IDF for keyword extraction is one of the major and one of the most unique parts of this research. Used to extract the keywords from each sentence of a text document. The purpose of the BERT encoder in this research is to encoding the sentences in a text document, in other words creating the embedding of sentences. These Embedding are then Further used to extract the Positional Embedding generated by BERT for every Topic and keyword extracted in previous layers. Then find mean of word embedding for each word from LSA topic modelling and name the mean value let say X. Do the same for each keyword from each sentence and store in Y(i). Once value of X and Y(i) is determined the next task is to use the Cosine similarity function to find compare each value of Y(i) vectors with X vector and store each value into Z(i), where i is the sentences number. After receiving the value of Z(i), the next task is to sort the value of Z(i). Then value of i corresponds to the top 5 maximum value of Z(i) is the desired sentences for the summary generation. These sentences are then combined to form required summary that contains the most relevant information about the long textual document.

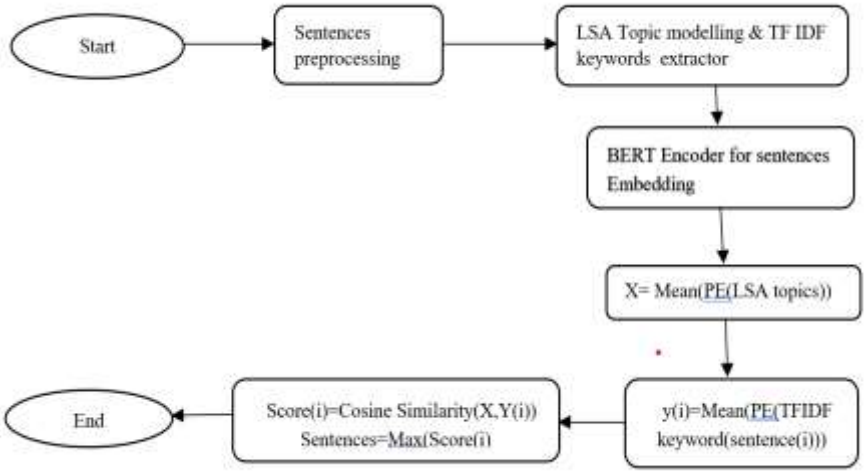

Fig: Latent semantic analysis

Extractive Text Summarization:

Pre-processing can be viewed as original text structured representation. It basically includes: 1) Identification of sentence boundary. In English, each sentence ends with full stop (dot). This is used to identify the sentence. 2) Elimination of stop-words. Words which do not give any relevant information or words with no semantic meaning are eliminated. 3) Morphological analysis – stem or radix of a word gives its semantics. The purpose of morphological analysis is to get the stem or radix of words. Lexical analysis transforms the source document into a set of tokens. Each sentence will be divided into number of tokens. Non-ASCII characters are removed since they do not carry implicit meaning. The final list of tokens is passed on to the next phase for further processing. Grouping of various inflected forms of a word is called lemmatization. The purpose of this is to analyse them as a single entity. In English, it is difficult to use a word in one single form. Often suitable equivalent forms of a word are used to grammatically avoid errors in a sentence. Once cosine similarity between vectors is identified, group similar sentences so that sentences are picked from each group. K means clustering partitions n observations into k clusters. Based on nearest mean value each observation fit into the cluster.

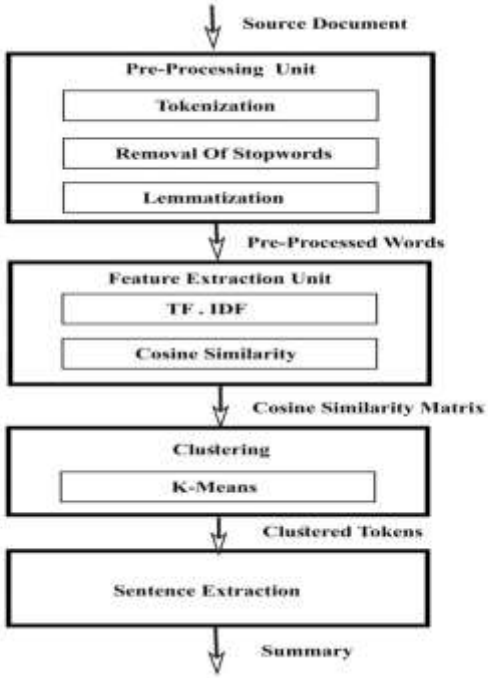

Fig: Extractive Summarization Method

K means clustering:

K-Means Clustering is an Unsupervised machine Learning algorithm, which groups the unlabeled dataset into different clusters. We applied K-Means clustering to divide a single large corpus into multiple sub-corpuses. Contrary to the traditional approaches, this approach used reduced corpus as a substitute for the large corpus. In this approach, the second stage KMC applied over grouped PRK of sub-corpuses. It further identifies relevant portions of SC and related it to a specific cluster. KMC produced unique sub-corpus clusters. After the second stage K-Means clustering, these clusters inter-related to each other in the same manner as followed to interlink the first-stage clusters. The first two layers consist of the clusters of FSC and SSC, joined in a path-oriented manner. The third layer contains keywords of the documents. The fourth layer has the GB summarized text. They generated the Query Specific Dynamic Corpus (QSDC) by processing the queries through these layers.

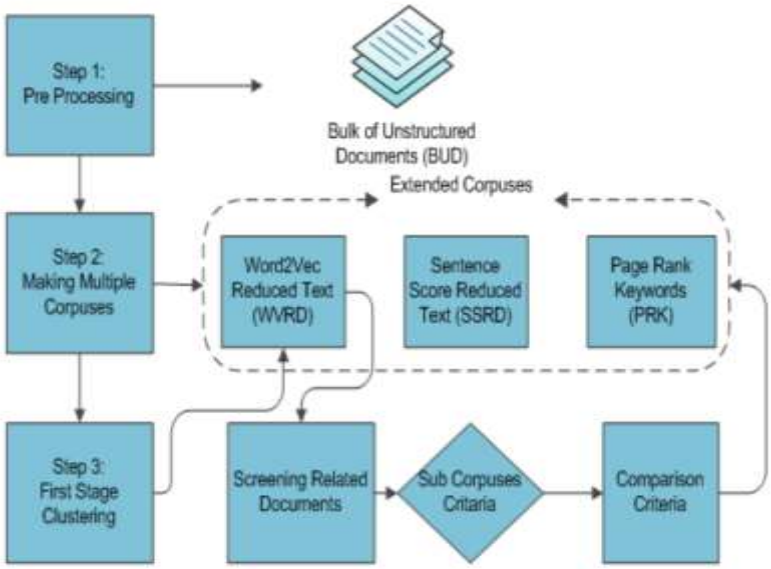

Fig1: Step by step hierarchy of FSC

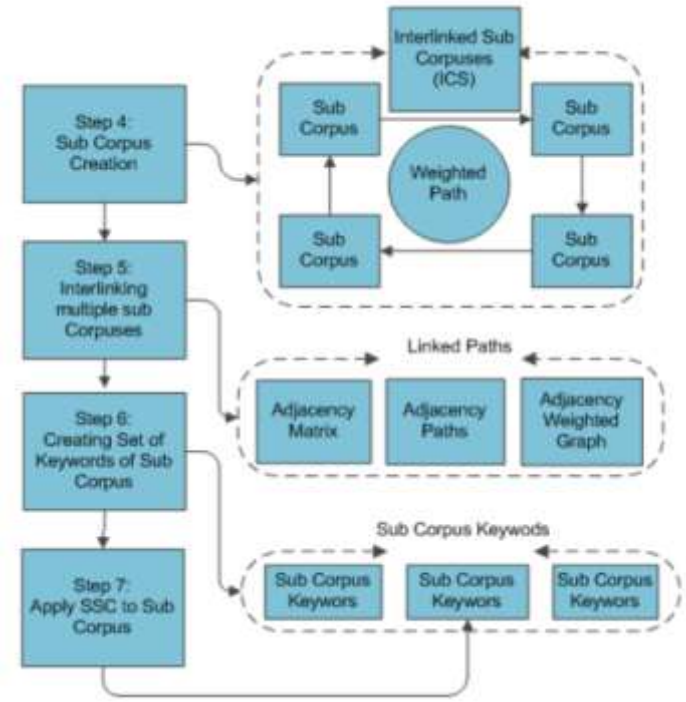

Fig2: Step by step processing of SSC.

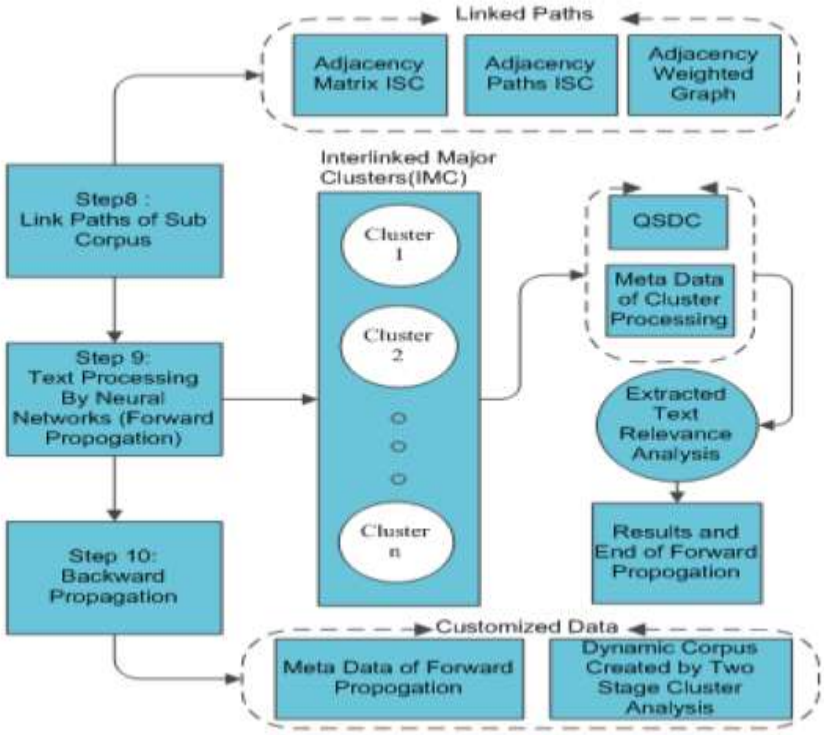

Fig3: Text processing over linked nodes of the corpus.

LFIP-SUM:

This Approach consists of a two-step procedure: document representation and representative sentence selection. To represent a document as a continuous vector, we use distributed vectors pre-trained by deep learning-based sentence embedding models. Next, ILP and PCA are used to evaluate the sentence importance score and select the representative sentences for the summary. They assume that a document consists of n sentences. As the sentences have sequential information, they can represent a document as a sequence of sentences as follows:

$D = [s_1, s_2, \ldots, s_n]$

where D denotes the document and sk refers to its *k*-th sentence. Thus, D_{basic}, the basic representation of D, becomes the following matrix:

$D_{basic} = [sv_1, sv_2, \ldots, sv_n]$

where $\mathbf{sv}_k$ denotes the pre-trained sentence vector of s_k. D_{basic} is a d × n matrix, where d is the embedding dimension. Therefore, we used positional encoding to effectively reflect sequential information. The positional encoding matrix PE is calculated as follows:

$PE_{(pos,2i)} = \sin(pos/10000^{2i/d})$,

$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d})$

PCA is a method for reducing high-dimensional data to lower dimensions. The principal components (PCs) removed by PCA are composed of a linear combination of the original variables. Hence, PCA has been widely used for dimensionality reduction because it is possible to explain the entire dataset through a few PCs.

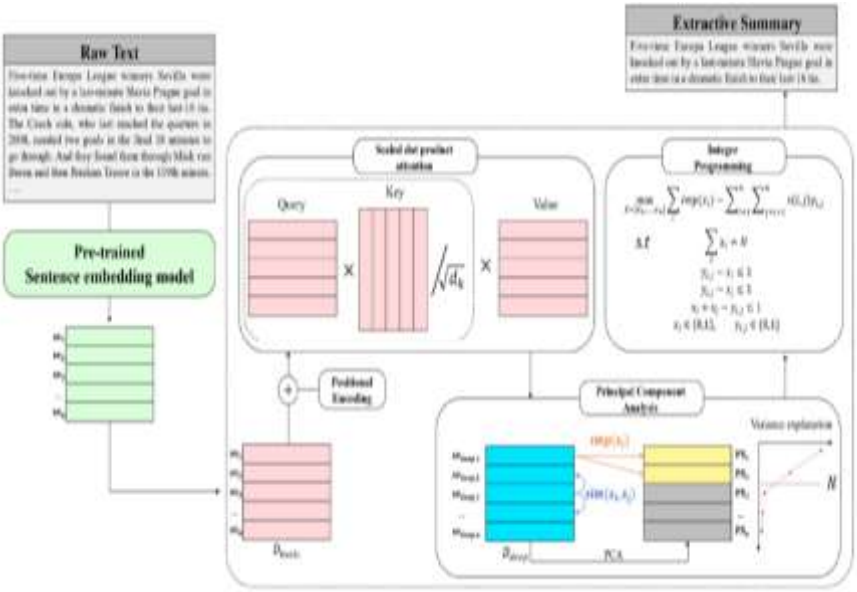

Fig: LFIP-SUM

RESULT :

The data in the table below shows the final outcome, which is used to summarize the text document. Applying NLP techniques to the text document after it has been prepared. Finally, ROUGE (Recall-oriented understudy for Gisting Evaluation) which is being displayed in Table. Fig shows the graphical representations of the ROUGE values.

ROUGE-1: It measures the overlapping unigrams, in other words means it measure overlapping words between original summary and human generated summary.

ROUGE-L: It measures the longest common sequence between the original and human generated summary.

S. No	Method	ROUGE-1	ROUGE-L
1	Bi-LSTM	25.37	29.03
2	LSA	45.33	37.33
3	Seq2Seq	27.14	30.89
4	ESN	29.91	30.91
5	K-means	33.24	35.16

Table: Different NLP methods with their ROUGE values

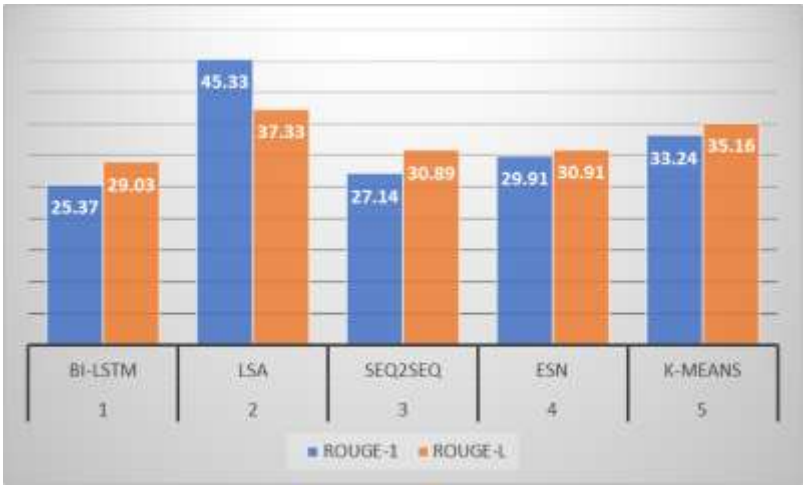

Fig: Analysis of various NLP methods

CONCLUSION

This paper examines the efficiency and accuracy of existing summarization systems. The summarizer systems in earlier stage mainly concentrate some simple statistical features of sentences and summarize only the technical articles. The above extractive summarization systems follow statistical, linguistic and heuristics methods. The statistical methods are tf method, tf-idf method, machine learning, lexical based, discourse based, cluster based, vector based, LSA based etc. The statistical methods follow supervised and unsupervised learning algorithms. The machine learning algorithm related models are generating coherent and cohesive summary but the algorithms are computationally complex and needs large storage capacity. These algorithms are overcome the redundancy in some extent and the systems are domain independent. Some systems follow the statistical and linguistic based methods. It also generates good summary but the linguistic analysis of source document required heavy machinery for language processing.

This paper covers extractive and abstractive summarization techniques. Summarization system should produce an effective summary in a short time with less redundancy having grammatically correct sentences. Both extractive and abstractive method yields good result according to the context in which they used. The reviewed literature opens up the challenging area for hybridization of these methods to produce informative, well compressed and readable summaries.

REFERENCES:

[1]. J. Jiang et al., "Enhancements of Attention-Based Bidirectional LSTM for Hybrid Automatic Text Summarization," in IEEE Access, vol. 9, pp. 123660-123671, 2021, doi: 10.1109/ACCESS.2021.3110143.

[2]. H. Gupta and M. Patel, "Method Of Text Summarization Using LSA and Sentence Based Topic Modelling With Bert," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 511-517, doi: 10.1109/ICAIS50930.2021.9395976.

[3]. K. Shetty and J. S. Kallimani, "Automatic extractive text summarization using K-means clustering," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 1-9, doi: 10.1109/ICEECCOT.2017.8284627.

[4]. M. Y. Saeed, M. Awais, R. Talib and M. Younas, "Unstructured Text Documents Summarization with Multi-Stage Clustering," in IEEE Access, vol. 8, pp. 212838-212854, 2020, doi: 10.1109/ACCESS.2020.3040506.

[5]. M. Jang and P. Kang, "Learning-Free Unsupervised Extractive Summarization Model, "in IEEE Access, vol. 9, pp. 14358-14368, 2021, doi: 10.1109/ACCESS.2021.3051237.

[6]. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, ''A decomposition-based multi-objective optimization approach for extractive multi-document text summarization,'' Appl. Soft Comput., vol. 91, Jun. 2020, Art. no. 106231.

[7]. A. Hernandez-Castaneda, R. A. Garcia-Hernandez, Y. Ledeneva, and C. E. Millan-Hernandez, ''Extractive automatic text summarization based on lexical-semantic keywords,'' IEEE Access, vol. 8, pp. 49896–49907, 2020.

[8]. Yang, X. Wang, Y. Lu, J. Lv, Y. Shen, and C. Li, ''Plausibilitypromoting generative adversarial network for abstractive text summarization with multi-task constraint,'' Inf. Sci., vol. 521, pp. 46–61, Jun. 2020.

[9]. B. Jang, M. Kim, G. Harerimana, S.-U. Kang, and J. W. Kim, ''Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism,'' Appl. Sci., vol. 10, no. 17, p. 5841, Aug. 2020.

[10]. Z. Li, Z. Peng, S. Tang, C. Zhang, and H. Ma, ''Text summarization method based on double attention pointer network,'' IEEE Access, vol. 8, pp. 11279–11288, 2020.