International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 05, 2020
ISSN: 1475-7192

# Abstractive Summarization of Text using Encoder-Decoder Based Architecture

[1]K.S. Agilan, [2]R. Aswathaman, [3]R. Harinisri, [4]M. Salomi

***Abstract*** *-The internet keeps bringing tons and tons of information to its users on a daily basis – reading everything can consume months or years and sometimes even decades. Access to this much information has helped us in several disciplines, but at times we are overwhelmed with information and end up getting confused. To solve this to an extent possible there are summarization techniques that automatically reduce the given text to a considerable size that can be easily studied. Such large text documents are quite impossible for humans to summarize and might end up useless since the scope for the information might change during the summarization process. This is where computers come into the scene. Computers are way better in handling large amount of data than humans since it can easily do repetitive tasks with accuracy and speed. Computers summarize and give us a comparatively small notes like document to simplify our workload. Though there are several techniques widely used for summarizing documents they can be widely classified into two categories - abstractive and extractive summarization. Many projects have been published on extraction summarization, however, it cannot provide summary close to human language. We try to provide summary close to human language using abstractive text summarization method. This project uses neural network models for abstractive summarization on long texts.*

***Keywords--*** *Abstractive* Summarization of Text using Encoder-Decoder Based Architecture

## I. INTRODUCTION

*. Overview*

Text summarization is the process of reducing the size of the given document into condensed form that keeps overall idea of the document. The techniques in text summarization are abstractive and extractive summarization. Summarization techniques requires NLP (Natural Language Processing) tools for summarizing the text documents. Summarization also requires statistical, linguistic and heuristic methods for ranking the sentences. Many methods have been used for summarization of text in various languages and various types.

With the unexpected growth of the Internet in the last decade, people often receive large amount of data from various sources, where in most scenarios it is text. For this reason, automatic text summarization is becoming an increasing necessity for diverse fields like internet search engines, business analytics, market review, academic purposes, etc., Automatic text summarization is generating a summary of a given document without any human involvement in the process

*[1] B.E,Dept.Of Computer Science and Engineering, KPR Institute of Engineering and technology,Coimbatore,TamilNadu*
*[2] B.E,Dept.Of Computer Science and Engineering, KPR Institute of Engineering and technology,Coimbatore,TamilNadu*
*[3] B.E,Dept.Of Computer Science and Engineering, KPR Institute of Engineering and technology,Coimbatore,TamilNadu*
*[4] Asst. Prof, Dept.Of Computer Science and Engineering, KPRInstitute of Engineering and technology,Coimbatore,TamilNadu*

**Received: 02 March 2020 | Revised: 21 April 2020 | Accepted:05 May 2020** 

This summarization is broadly classified into two classes — extractive and abstractive summarization. Extractive summarization use exact phrases and words from the original document depending on the level of its importance, whereas abstractive summarization produces a summary using sentences or phrases that might not be a part of the given source document.

Building an abstractive summary is a difficult method and also involves complex language modeling structures. We are focusing on abstractive text summarization in this project using a neural network.

*Extractive* Summarization

Here, content is taken from the original data, but the extracted content is not changed in any way. Examples of extracted content include key-words that can be used to "tag" or index the text document, or key sentences that collectively contribute to a concise summary. For text, extraction is analogy to the process of skimming, where the summary consist of headings and subheadings, figures, the first and last paragraphs of the section, and optionally the first and last sentences in a paragraph are usually given more importance while summarizing(Jarble, 2020).

*Abstractive Summarization*

Abstractive summarization has mostly been applied for text. Abstractive methods usually build an internal semantic representation of the original text, and then use the said representation to create a summary closer to how a human might express. Abstractive summarization transforms the content by paraphrasing said sections of the source document, to condense the text more strongly than extractive summarization. Those kind of transformation, is computationally challenging than extractive summarization, it involves both NLP and an understanding of the domain of the original text in cases where the original document relates to a special field of knowledge. Most summarization systems are extractive in nature(Jarble, 2020).

To understand the difference between them better, let's consider the following paragraph-"Educators all around the world have fears about instituting large systemic changes, and sometimes those fears are well grounded. Even though, we cannot afford to ignore the possibilities that AI offers us for drastically improving the student learning experience. People need to comprehend the role of AI in improving the role of education in the years to come. The resistance faced by this new technology should decrease because AI can not only help the teachers be more productive, but also make them more responsive towards the needs of the students."

The extractive summary of this would be-"However, we cannot afford to ignore the possibilities that AI offers us for dramatically improving the student learning experience. The resistance faced by this new technology has to decrease because AI can not only help the teachers be more productive, but also make them more responsive towards the needs of the students."

The abstractive summary of the same paragraph would be-"Educators will have to overcome their fears and reduce the resistance faced by AI to implement it in the education system because of the opportunities it offers."

By comparison we observe that the abstractive summary is much shorter and to the point than the extractive summary.

Recurrent Neural Network(RNN) is a type of Neural Network in which **the** output of the previous step is fed to **the** next step as input much like a feedback loop. In traditional neural networks, all the inputs and outputs independent, but when it is required to predict the consequent words in a sentence, the previous words are necessary

and henceit is necessary to remember the previous words in the exact order. Thus RNN's were created, which solved the feedback issue with the help of Hidden layers.

The most important feature of RNN is the Hidden state**,** which remembers some information about a sequence and the order. RNN's have a "memory" which remembers all information on what has been calculated. It uses the same parameters for each input as it does the same set of tasks on all the inputs or hidden layers repetitively to produce the output. This reduces the complexity of parameters to very low, not like other neural networks.

It is observed that Recurrent Neural Network based approaches are promising and give hope to solving Abstractive text summarization problem which had been largely unsolved till now. But the problems are with the metric and lack of dataset which challenge the scalability and generalizing to multi-sentence summarization.

The encoder-decoder based architecture for RNN's is a standard neural machine translation method that oposes and in some cases performs better than classical statistical machine translation methods.This architecture is very new, having only been pioneered recently, but Google's service called Google translate has adopted it as the core technology.

The key benefits of the approach are the ability to train an end-to-end model directly on source and target sentences and also the ability to handle variable length input and output sequences of text used. Encoder Decoder RNN model is used in order to solve all the limitations NLP for text summarization faces for getting a short and accurate summary. It is a much more intelligent approach.

Thus the summarization done using a encoder decoder RNN system will be more effective than other techniques.

## II. BACKGROUND STUDY

***Text summarization using TF-IDF***

The Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical pointer which reflects how important a word is to a document. This method is often used as a factor in information retrieval and text mining. TF-IDF is used to stop word filtering in text summarization and categorical application. The value of TF-IDF increases proportionally to the number of times that a word is present in a document, but is offset by the frequency of words in the corpus, which controls the fact that some words are more common than others in the language. Here the term frequency means the calculated raw frequency of a term in a document. The term regarding inverse document frequency is a measure of whether the term is rare or common across all documents which can be obtained by dividing the total number of documents involved by the number of documents containing the term only.

***Query bases and generic summarization***

In query based text summarization the scoring of the sentences of the document is done based on the frequency counts of words or phrases. High scores are given to the sentences in the document containing the query phrases, than the ones with query words. The sentences with highest scores are the ones that are extracted for the output summary together with their structural context. Parts of text can be taken from different sections wherever they are. The resulting summary is the culmination of such extracts. In the sentence extraction algorithm, whenever a sentence is selected to be included in the summary, some of the topics that are relatable are also selected.

Summaries that are based on queries are biased as they do not provide the necessary overall review of the source document. They interact with the queries given by the users only hence not suitable for content overview. A best generic summary considers the main topics of the documents and tries to minimize the redundancy as much as possible.

***Bayesian classifier***

In a training set of documents with selected document extracts, we develop a classification function that estimates the probability that a given sentence is included in an extract. New extracts can be generated by ranking the sentences according to the obtained probability and selecting a user-specified number of the top scoring ones. For each sentence s we compute the probability it will be included in the summary S calculated with the k features fj given; j = 1…k, which can be expressed using Bayes' rule as follows

$$P(s \epsilon S \mid f1, f2, \ldots, fk) = P(f1, f2, \ldots, fk \mid s \epsilon S)\ P(s \epsilon S)\ P(f1, f2, \ldots, fk)$$

Equation 1 Bayes' rule

Assuming statistical independence of the features:

$$P(s \epsilon S \mid f1,2, \ldots, fk) = \prod P(fj \mid s \epsilon S)\ P(s \epsilon S)\ kj=1 \prod P(fj)\ kj=1$$

Equation 2 Bayes with statistical independence

***Fuzzy logic based summarization***

Here, Fuzzy Logic rule and fuzzy logic set are used to extract the important sentences based on the features. Fuzzy logic techniques provide decision-support systems and expert systems with strong reasoning capabilities. The fuzzy sets can be used to design a new co reference algorithm which captures this uncertainty of the words in an explicit way and allows users to define varying degrees of reference. People also use Fuzzy Logic for scoring sentences after feature selection and pre-processing steps. The model uses eight features for text summarization: title word, sentence length, sentence position, numerical data, thematic words, sentence to sentence similarity, term weight and Proper Nouns.

Fuzzy Logic System includes four components: the fuzzifier, the Inference Engine(IE), the Defuzzifier and Knowledge Base. In the fuzzifier, inputs are translated into linguistic values and it uses a membership function to be used to the input linguistic variables. After fuzzification, the inference engine refers to the rule base using fuzzy IFTHEN rules to derive the linguistic values.

Disadvantages of existing systems

- The important features of the words are not utilized optimally
- The sentiment of the text is not totally understood by these mechanisms
- The productivity is low and the accuracy of the summarized text is not in par with neural network based mechanisms.

**Received: 02 March 2020 | Revised: 21 April 2020 | Accepted:05 May 2020** 

# III. IMPLEMENTATION

### *Approach*

The approach followed for summarization is based on LSTM based encoder decoder architecture with attention mechanism

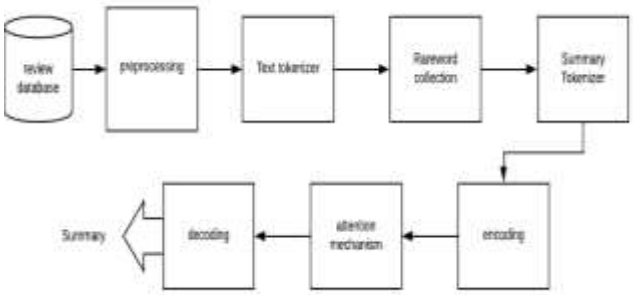


**Figure 1:** LSTM based encoder decoder architecture

### *About the data*

The data which we use for training the model is the "Amazon fine food review" data. This dataset consists of reviews of foods from amazon. The data is obtained over a period of more than 10 years, including all ~5,00,000 reviews 2012. Reviews contain product and user information their ratings, and a text review. It also includes reviews from all other Amazon organization's categories. We use our algorithm to build the Summarizer to shrink the review's size to get the gist of the customers review.

```
[ ] data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 393565 entries, 0 to 568453
Data columns (total 10 columns):
Id                      393565 non-null int64
ProductId               393565 non-null object
UserId                  393565 non-null object
ProfileName             393565 non-null object
HelpfulnessNumerator    393565 non-null int64
HelpfulnessDenominator  393565 non-null int64
Score                   393565 non-null int64
Time                    393565 non-null int64
Summary                 393565 non-null object
Text                    393565 non-null object
dtypes: int64(5), object(5)
memory usage: 33.0+ MB
```

**Figure 2:** Input data

### *Preprocesing*

Preprocessing is the process of converting raw data into useable data. Raw data cannot be processed as it is because it contains many factors which affect the accuracy of the model. The data in our data set contains stop words, empty rows, HTML tags, punctuations and special characters in it. These are rectified during preprocessing.

***Tokeniation***

Tokenization is the process of breaking up a sequence of strings into parts like words, keywords, phrases, symbols and other elements called tokens. Tokens can be words, phrases or whole sentences. In the process of tokenization, characters including punctuation marks are discarded. The tokens become the input for the summarizer. Tokens and words are separated by whitespace, punctuation marks or line breaks. White spaces and punctuation marks need not be included depending on the need. All characters within continuous strings are part of the token. Tokens are made up of all alpha characters, alphanumeric characters or numeric characters only.

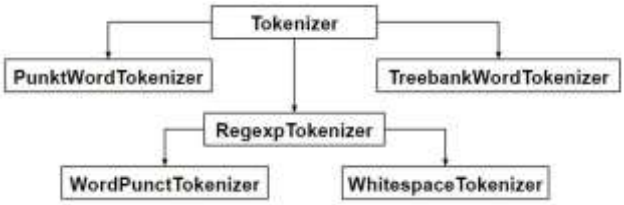


**Figure 3:** Tokenizer architecture

***Rare word collection***

Rare words are certain words which are a part of the English vocabulary but are not used very often. These words have a meaning but are not known to the computer as they are not used in the corpus so we have to take all the rare words and use it for training the model. These words cannot be discarded because they have meaning and can also denote the user's sentiment.

```
[ ] thresh=6

    cnt=0
    tot_cnt=0
    freq=0
    tot_freq=0

    for key,value in y_tokenizer.word_counts.items():
        tot_cnt=tot_cnt+1
        tot_freq=tot_freq+value
        if(value<thresh):
            cnt=cnt+1
            freq=freq+value

    print("% of rare words in vocabulary:",(cnt/tot_cnt)*100)
    print("Total Coverage of rare words:",(freq/tot_freq)*100)
```

**Figure 4 :** Rare word collection (occurs less than 6 times in this case)

***Encoding and decoding***

An encoder is a type of network which takes the input provided, and gives the output as either a feature map or a vector or even a tensor. The feature vectors hold the information on the features which are used to represent the input. The decoder is a network which has the same structure as that of the encoder. The difference is that the decoder possesses the orientation which is opposite of the encoder. The decoder takes a feature vector and provides a match that is the closest to the actual input.

The encoders and the decoders are trained in a complementary way. The encoders and decoder mechanism is unsupervised in nature. The optimizer will try to streamline both the encoder and decoder to reduce the loss. After

the training process, The encoder and decoder provides us with a format to work with the data. The encoding and decoding need not be manually done hence reducing the stack for the user.

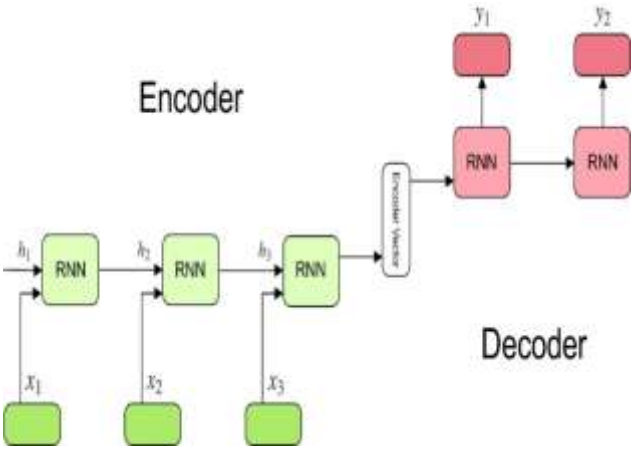


**Figure 5:** Encoder Decoder architecture

***Training***

Neural networks consist of neurons which are connected to one another, every separate connection of the neural network is given a weight which gives the importance of each of the relationship of the neuron when they are multiplied by their input values.

Each neuron has an activation function that is used to determine the output of that neuron. The activation function is used for introducing the concept of non-linearity to the modeling of that neural network. There are lots of types of activation functions here, sigmoid function is used.

The training process is said to be the go and return mechanism where go represents forward propagation and return represents the backward propagation process

The first phase which is forward propagation takes place when the neural network is allowed to use the training data. The training data passes throughout entire neural network and their class labels are calculated. This process is essentially passing the input data through the network so that the neurons can process their transformation of the information obtained from the previous layer and sending the processed information to the neurons in the next layer. After the data passes through all the layers in the neural network, and all the neurons in the network done their calculations, the final layer obtains the class labels as a result

As the next step we obtain **the** loss function. Loss function is used for loss estimation and for comparing how good the obtained prediction result was in comparison with the training class labels. An Ideal Network has its cost as zero, which means that there is no difference between the expected and the actual values. In hope of finding the ideal network the values in the connections of the neurons will be adjusted gradually.

**Received: 02 March 2020 | Revised: 21 April 2020 | Accepted:05 May 2020** 

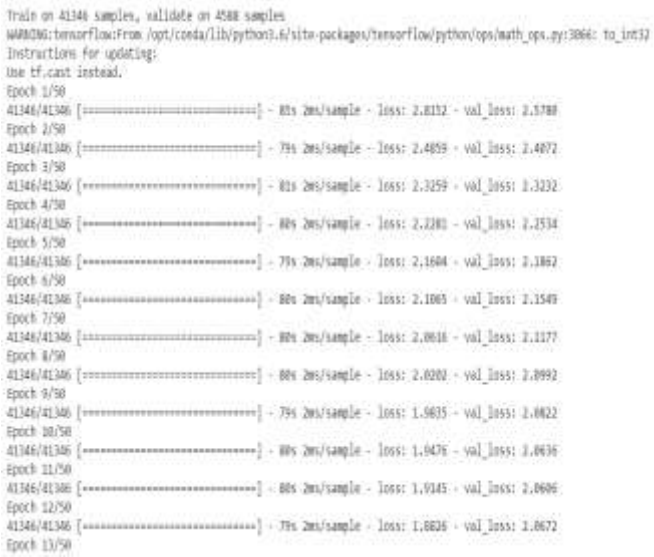


**Figure 6** : Monitoring validation loass during training

After the loss function is calculated, the information in the last layer is propagated backwards. The information of loss and the adjustments are propagated to all neurons in the inner layers which are responsible for output.

Even though the data is propagated backwards the information on the loss is not reflected to all the neurons responsible. Therefore we use repeated back propagation help information to reach all the neurons that are involved in their contribution to the total loss. Some weights are adjusted in such a way that it helps to improve the loss. A technique called **gradient descent is used to** reduce the loss. In gradient Descent we change the weights in small doses by calculating the derivative of the loss function. This derivative helps us to find out in which direction reduce the loss to obtain the global minimum. This whole process is done batches of data in consecutive iterations. These iterations are called epochs. Epochs are the iteration of all the data passed to the network in each iteration.

Accuracy

Accuracy is the measure of how well the given neural network performs. It is the measure of how well the given input is utilized to obtain the required output. The measure for not obtaining the optimal accuracy is known as loss. Loss is measured in two ways

- Training loss
- Validation loss

Training loss is the inaccuracy of the neural network model over the training data whereas the validation loss is inaccuracy if the model over a separate data which is known as validation data. This validation data set is used solely for finding the performance of the model

The rubrics for measurement of performance of the model are:

- The validation loss ought to be similar to that of training loss but slightly higher in value. Training must be done as long as the validation loss is lower than or in some cases equal to the training loss
- If the training loss reduces but there is no increase in the validation loss then we must keep on training the model.
- If the validation loss starts increasing then we must stop the training or else we will get overfitting

If accuracy is not acceptable then we must experiment with the following parameters to find the correct fit

- Should we add more data?
- Do we add more data augmentations?
- Should we try different data augmentations?
- Is there a problem with data?
- Should we try a different architecture type?

The accuracy of training and accuracy of validation both needs to be high for a good prediction. The training and the validation loss for the model is :

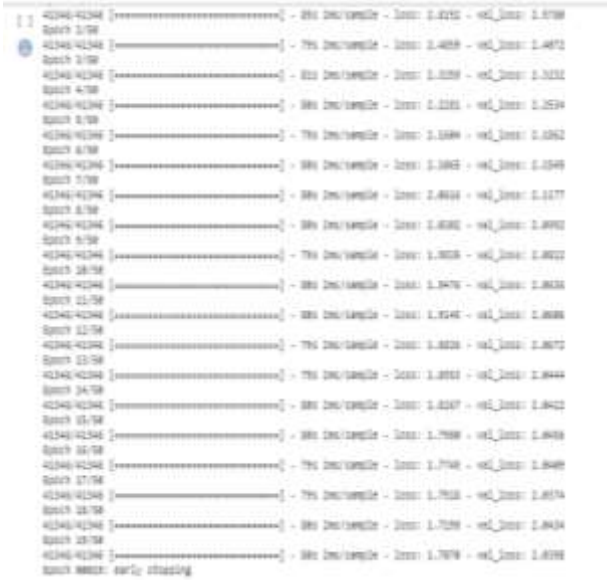

**Figure 7:** Training and validation loss

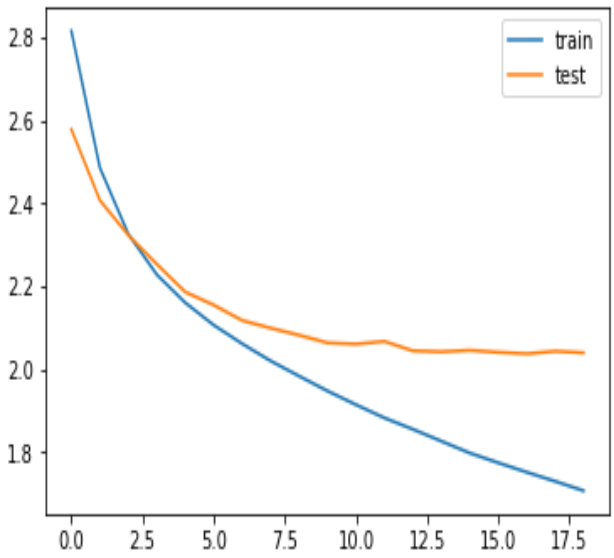


**Figure 8:** Train vs Test

Using the above plot, we can identify that validation loss has increased after the 17th epoch for 2 consecutive epochs. therefore, the training is stopped at the 19th epoch. If we train for a longer duration the training loss may decrease but the validation loss for our model will increase which will lead to overfitting.

K. Output

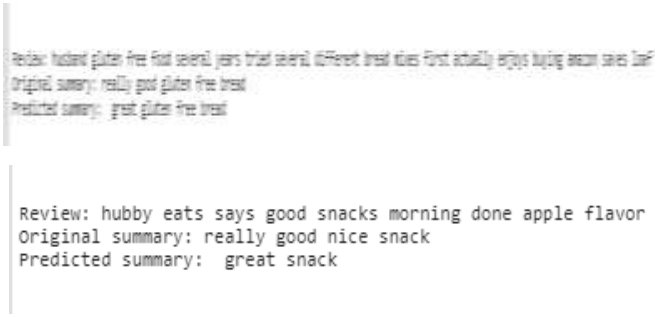


**Figure 9:** Output

The screenshot provides the output our summarization model , the maximum size of our predicted summaries is 4 and our summaries give the meaning of the review in very short form it also portrays the sentiment of the reviewer.

## IV. CONCLUSION

The advent of the internet has created a huge amount data which in turn requires tremendous effect to summarize the overall view on a certain topic that could support a decision. A short summary is used to convey the essence of the document it also helps in finding the relevant information about anything very quickly. Document summarization paves the way to cluster documents and present summaries in a concise manner. In essence, summarization is meant to help us consume relevant information faster and thus reduce the time spent in decision making. The above abstractive summarization mechanism provides an effective means for understanding the essence of the text

## V. FUTURE SCOPE

The software can be implemented with a GUI and can be made into a user interactive system for text summarization. The summary generator can also be developed as an autonomous system and they can be used in several fields wherever decision making becomes important. The summarizer can also be made to be a multi document summarization system.

## REFERENCES

1. R. Nallapati, F. Zhai, and Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. arXiv preprint arXiv:1611.04230, 2016
2. Das, D. and Martins, A. F. (2007). A survey on automatic text summarization.

3. Ramesh Nallapati, Bowen Zhou, CaglarGulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023, 2016
4. Gambhir, M. and Gupta, V. (2017). Recent automatic text summarization techniques: a survey. Artificial Intelligence Review, 47(1):1–66
5. N. Bhatia and A. Jaiswal, "Automatic text summarization and it's methods-a review," in Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference. IEEE, 2016, pp. 65–72Mani and M. T. Maybury, Advances in automatic text summarization. MIT press, 1999.
6. S.A. Babar, P. D. Patil, "Improving Performance of Text Summarization", International Conference on Information and Communication Technologies, ICICT, 2014.
7. Sherry, P. Bhatia, "A Survey to Automatic Summarization Techniques", International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015.
8. S. Karmakar, T. Lad, H. Chothani, A Review Paper on Extractive Techniques of Text Summarization, International Research Journal of Computer Science (IRJCS) Issue 1, Volume 2, 2015.
9. F. C. Pembe and T. Güngör, "Automated Query-biased and Structure-preserving Text Summarization on Web Documents," in Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul, June 2007.
10. G. Yihong, X. Liu. "Generic text summarization using relevance measure and latent semantic analysis." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2001
11. S. Alfayoumy, J. Thoppil, A Survey of Unstructured Text Summarization Techniques, International Journal of Advanced Computer Science and Applications, Vol. 5, No. 4, 2014.
12. L. Suanmali, N. Salim, M. S. Binwahlan, "Fuzzy Logic Based Method for Improving Text Summarization", International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009.
13. P.Priya, G. and K Duraiswamy, "An Approach for Text Summarization Using Deep Learning Algorithm", Journal of Computer Science, 19, 2014.
14. Amy J.C. Trappey, Charles V. Trappey, "An R&D knowledge management method for patent document summarization" Industrial Management & Data Systems, vol.108. pp. 245-257. 2008.