# Extractive Text Summarization using Deep Natural Language Fuzzy Processing

**Neelima G, Veeramanickam M.R.M, Sergey Gorbachev, Sandip A. Kale**

***Abstract— Text summarization is most trending research areas in a modern context. The main aim of this project is to reduce text size while preserving the information underlying into it. In summary construction level, in general, given complex task which are basically will involve with deep natural language fuzzy processing methodologies. In general, an extractive based summary method is the very simple original text of subset of which will not guarantee as best narrative coherence output, because they are most conveniently representing an approximate summarized content from given text-based only on relevance judgment. In an automatic process of fuzzy summarization which is divided into the following steps: Pre-processing (sentence segmentation, tokenization, stop words removal), Feature Extraction, Sentence Scoring, Sentence Ranking and Summary Extraction.***

***Index Terms— Natural Language Fuzzy Processing, Text Summarization, Tokenization, Naive-Bayes.***

## I. INTRODUCTION

In today's world, we can get information everywhere. It has never been this much accessed in the past until now. With the growth of the internet, we entered the era of information technology. Information and data are produced every day at a massive rate. There are over 1.8 billion websites currently on the internet. We cannot imagine how much information these many websites contain. But this information doesn't guarantee the usefulness for the user. Some information contained in such massive websites might be of less use to the user, some information might be duplicated, and some might contain noise in it [1] [2]. Noise refers to the incompleteness of documents, missing characters or use of unwanted characters, and so on. The given input can retrieve the relevant and essential point's information from a document, its summarization phase playing a vital role. In the computing field teaching and learning using personalized learning is most required platform with social learning, internet of things and ANN [3][4]. This summarization model helps to deployed in e-learning platform.

The communication of human between each other will be done using the Natural language. By using the huge amount of data, the process of communication will be happening and from that useful information will be occur, by that information it allows the computer to make more communicate with the human. NLP (Natural Language Processing) refers to techniques and methods involved in automatic manipulation of natural language [5]. Human-computer interaction, machine learning, information summarization and some more are using the Natural language [6].

Summary refers to a text, or a paragraph or a document having less size than the original text, or a paragraph or a document and containing the most important meaning from that text, or a paragraph or document. It is impractical to construct a summary of each and every document found in today's world manually. Instead, we can automate the process of constructing a summary of documents so that only selected documents can be summarized. We can construct a summary of two kinds of documents, i.e. single document and multiple documents. The given single input document can do summarization which refers to the generating process for summary output from a one document, but in case of multi-document summarization level the process generate single output summary with help on using multiple given input documents. There are different ways of summarization for a document [7]. E-learning used for notes sharing to help of personalized model using internet of things and summarization [8] [9].

Extractive summarization: The basic approach is to extract document parts as per deemed interest for summarization on certain metric like example: inverse-document frequency mentioned as the tf - IDF: this so called often as a weighting factor, this factor value is increased proportional to the number of times a word appears in the document. The weight of terms diminishes based on Inverse document frequency factors which occurred frequently allow to increase the weight terms occurs rare manner, for example, the word "the", "and", "a" appear most frequently but they aren't helpful in giving the required information about the document) [11] [12].

Original Text: Alice and Bob went by the train for visiting the zoo and saw animals like a baby giraffe, a lion, and a group of birds in the colorful tropical area.

Extractive Summary: Alice and Bob visited the zoo. Saw a flock of birds in a group.Many of the times we can notice about the extractive summarization constraint which makes the output summary as an awkward sentence or grammatically strange one.

Abstractive summarization: Second approach for simplifying summarize as similar to humans doing, which is different from imposing extractive constraint and then allow for re-phrasings the content.

Abstractive summary: Alice, Bob visited the zoo and saw birds and animals. For another example, if the case summary

**Revised Manuscript Received on April 12, 2019.**

**Neelima G,** Vignan's Institute of Information Technology, Visakhapatnam, Andhra Pradesh, India.

**Veeramanickam M.R.M,** Vignan's Institute of Information Technology, Visakhapatnam, Andhra Pradesh, India.

**Sergey Gorbachev,** Candidate of Technical Sciences, National research Tomsk State University, Russia.

**Sandip A. Kale,** Trinity College of Engineering and Research, Savitribai Phule Pune University, Pune, India.

using text not listed from the given input text, then maintain a number of the words for similar information.This project focuses on generating a summary using extractive text summarization method [13] [14].

## II. PROPOSED PROBLEM: SOLUTION

In order to generate a fairly accurate summary, we have to be able to select the most relevant and the most important sentences from a document using effective measures. We have taken a Naive-Bayes approach for determining the important sentences from a given document. In this approach, we have taken the following steps for getting results.

### *A. Tokenization:*

It used to divide given input sentence into number of word chunk. Basically, this tokenization model is to perform tasks in NLP-natural language procesing pipeline way. This can help to perform at two different levels: word-level and then sentence-level. First one is Word-level tokenization which returns a group of words in a given sentence.
Example: I feel uncertain. => [‘I’ ‘feel’ ‘uncertain’]

In case of sentence-level tokenization which returns a chunk of sentences from is given document input.

### *B. Pre-processing of document*

As there are unlimited sources of information in the modern world, the input document that we receive may not be in proper English format i.e. it might contain noise in it. The noise might be in the form of special characters, unwanted spaces, new line characters, stop words, etc. Thus, we perform the following operations in the input file to obtain only an informative part of the document:
Step 1: Remove all new line and carriage return characters
Step 2: Remove all brackets and special symbols with numbers
Step 3: Remove all commas, extra spaces and duplicate sentences

### *C. Removal of stop words*

In these steps removing all stop words from given input as per natural language. These stop word which does not give any meaningful information for the given context. For illustration, as if we are developing an emotion detection collection of words like "is", "am", and "the" which do not convey any information relate to that emotion.

For instance, in this given sentence "I am feeling sad today", the beginning two words "I" and "am" this can be removed because these words does not providing emotion-related message. Even though, "I" words is required as per importance of other reasons, to know who is feeling sad like identification. So there is not specific common universal stop word list which helps to remove it. Hence, it’s totally depends upon user defined application [15] [16].

In natural Language processing, for each word processing is required. By that it is suitable that has only those processed words in our performed text that are important in a context, by that we can save time in processing and results in a more robust NLP engine.

### *D. Stemming*

Stem or root of a word refers to the main word from which other form or derivatives of that word can be formed. In this step the process of suffixes strips from given input words for normalizing it and reduces to their non-changing portion. For illustration, this is similar context like "computational", "computed", "computing" will lead to, result in a single word as "compute" hence it’s the non-changing part of the word in given context inputs. In stemming functional working on one word at the time and will not take care of word context into the account. But however, example, likes "compute" which do have semantic information context towards stemming. One sentence may contain a word like 'playing', and the next sentence may contain a word like 'plays'. Generally, the main concept of those two sentences can be assumed to be related to some game, as the stem of 'playing' and 'plays' is 'play'. Thus, stemming help in determining if a sentence is relatively important based on the frequency of stem of words it contains [17].

### *E. Fuzzy Processing and Extraction of important sentences*

There must be some measure to assess the sentences importance in the document. For extracting the important sentences from the document, the following calculations are done:

Step 1: Calculate the frequency of all the words from the pre-processed text.

Step 2: Calculate the weighted frequency of each word by dividing the word frequency with the maximum frequency.

Step 3: Tokenize all the important sentences in the given input.

Step 4: Calculate the sentence score by adding the weighted frequency of words contained in that sentence.

Step 5: Sort the tokenized list of sentences based on their scores in descending order.

Step 6: Extract ‘n’ sentences from the tokenized list [18].

## III. IMPLEMENTATION OUTPUT: EVALUATION & RESULTS

The sentences extracted by this text summarizer are completely based on their individual scores which are calculated by adding the weighted frequencies of the words present in it. The sentences with higher scores are generally found to be one of the important sentences in most cases. However, sometimes, sentences with longer length will have greater scores which reduce the quality of the text summarizer. And in situations where sentences are selected from a random part of the document doesn’t always give a meaningful combination of sentences. But the general idea of any document can be retrieved using this text summarization technique.
Experimental Analysis:The first displayed screenshot is user interface about the main screen of the implemented application outputs.


**Figure 1: Main user interface when the user runs the application**

The above Figure 1 shows the user interface by which input file can be given through the document or by the web link by clicking the summarize the text or summarize the existing file.

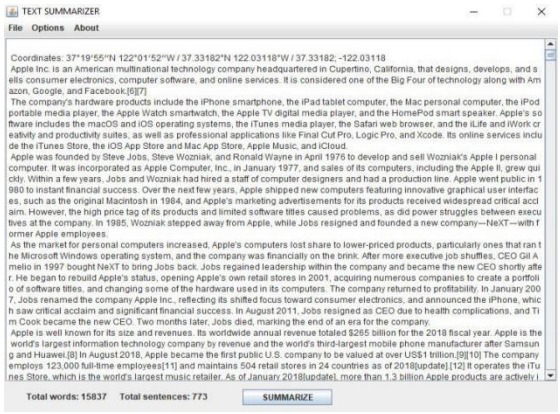


**Figure 2: Inbuilt text editor where all the contents of file or webpage are displayed**

The above Figure 2 shows the text after uploading the input, which was done by the above-said figure 1.

This Figure 3 shows the summarize setting screen where we can get the summarize selected sentences by assigning the number. By that those numbers of sentences will be displayed from the original input file.

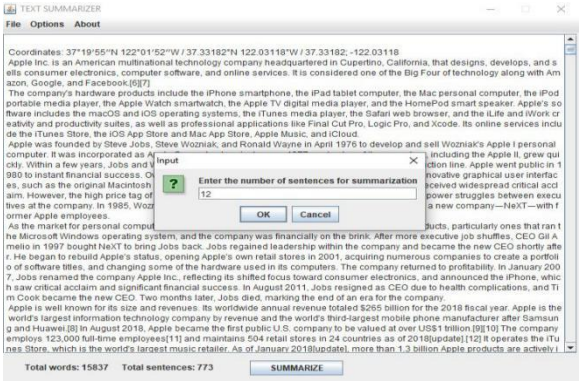


**Figure 3: summarize settings screen (For setting the number of sentences as the summary)**

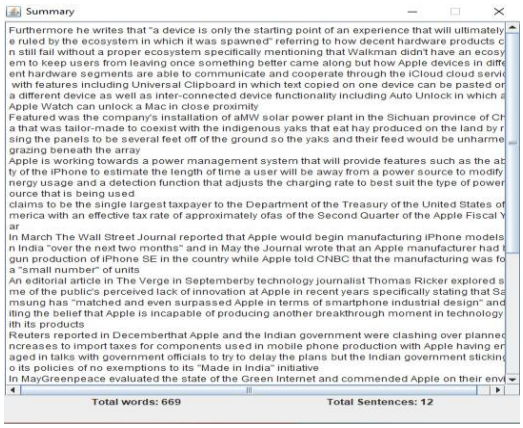


**Figure 4: Summary of a given text document or webpage**

This figure 4 shows the summary of the document with the total number of words and total sentences.

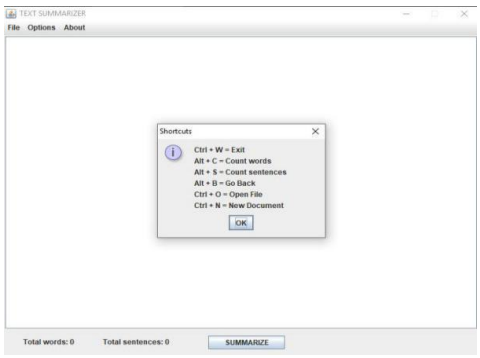


**Figure 5: App shortcuts for users**

The above figure 5 shows the shortcuts by which user can manipulate or process the operations by clicking the respective shortcut keys which were one of the contributions of the paper.

Following are the shortcuts created:

Ctrl+w=Exit, Alt+c=Count words, Alt+s= Count Sentences

Alt+b= Go back, Ctrl+0= Open file, Ctrl+0= New document

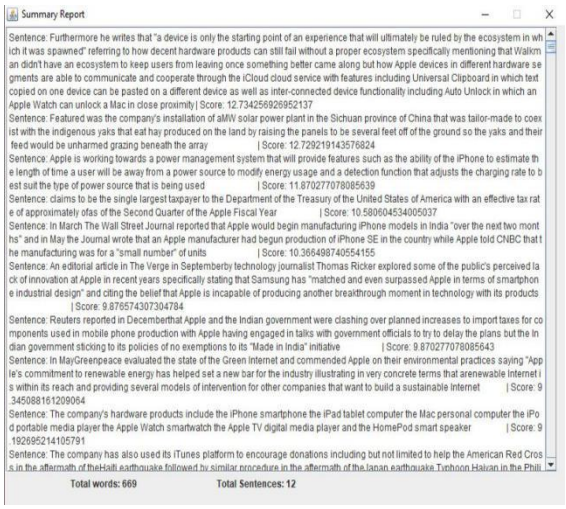


**Figure 6: Summary report with scores of each sentence contained in the document or webpage**

It shows the summary report with scores of the document with a total number of words and total sentences.

Future research should consider the potential effects of extractive summarization like the selection of sentences from different parts of a document which sometimes doesn't relate to each other. Future summarizer should also add support to work with other file formats like doc and pdf. We also believe that future research should work on the optimization of important sentences selection algorithm and sentence scoring for getting better results.

## IV. CONCLUSION

Hence concluded that automatic text fuzzy summarization based on deep natural language fuzzy processing methodologies is really useful for summarizing large documents and the contents of any web page. It saves valuable time and makes information gathering easier by allowing users to collect and work only relatively the most important information from any given input after summarizing.

## ACKNOWLEDGMENT

The paper was supported by the Russian Foundation for Basic Research, grant № 16-29-12858.

## REFERENCES

1. Merchant, K., & Pande, Y. (2018, September). NLP Based Latent Semantic Analysis for Legal Text Summarization. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1803-1807). IEEE.
2. Zopf, M., Mencía, E. L., & Fürnkranz, J. (2018, June). Which Scores to Predict in Sentence Regression for Text Summarization?. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 1782-1791).
3. M.R.M. Veeramanickam, N. Radhika, 'A Smart E-Learning System for Social Networking', International Journal of Electrical and Computer Engineering (IJECE) Vol. 4, No. 3, June 2014, pp. 447~455 ISSN: 2088-8708.
4. Veeramanickam MRM, Mohanapriya M, et al., 'Map-reduce framework based cluster architecture for academic student's performance prediction using cumulative dragonfly based neural network.' Cluster Computing. 2018. https://doi.org/10.1007/ s10586-017-1553-5
5. Aggarwal, C. C. (2018). Text Summarization. In Machine Learning for Text (pp. 361-380). Springer, Cham.
6. Peter Liu and Xin Pan, "Text summarization with TensorFlow"https://ai.googleblog.com/2016/08/text-summarization-with-tensorflow.html retrieved on 10th May 2019.
7. "User Review Data Set for Sentiment Analysis, Opinion Mining and Summarization" http://textanalytics101.rxnlp.com/2011/07/user-review-datasets_20.html, retrieved on 10th May 2019.
8. M.R.M.Veeramanickam, Dr M. Mohanapriya, "IOT enabled Futurus Smart Campus with effective E-Learning: i-Campus", GSTF Journal of Engineering Technology (JET), Volume 3, Issue 4, pp.81-87, April 2016.
9. M.R.M. Veeramanickam, M. Mohanapriya, et al., 'Research Study on Applications of Artificial Neural Networks and E-Learning Personalization'. IJCET, 8(8), 2017.
10. Zopf, M., Botschen, T., Falke, T., Heinzerling, B., Marasovic, A., Mihaylov, T.,& Frank, A. (2018, October). What's Important in a Text? An Extensive Evaluation of Linguistic Annotations for Summarization. In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 272-277). IEEE.
11. Jason Brownlee, "A Gentle Introduction to Text Summarization", https://machinelearningmastery.com/gentle-introduction-text-summarization, retrieved on 10th May 2019.
12. Kanapala, A., Pal, S., & Pamula, R. (2019). Text summarization from legal documents: a survey. Artificial Intelligence Review, 51(3), 371-402.
13. Jo, T. (2019). Text Summarization. In Text Mining (pp. 271-294). Springer, Cham.
14. Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023.
15. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268.
16. Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., & Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. arXiv preprint arXiv:1805.03616.
17. Khan, A., Salim, N., Farman, H., Khan, M., Jan, B., Ahmad, A., & Paul, A. (2018). Abstractive text summarization based on improved semantic graph approach. International Journal of Parallel Programming, 46(5), 992-1016.
18. Gorbachev S.V. (2017). A method to identify the significance of technological features in a logically transparent neuro-fuzzy models. Advanced Engineering Research and Applications, Ed. Hongseok Choi, India Research Publication, New Dehli, India,. Chapter 16. – pp.243-254