Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)
IEEE Xplore Part Number:CFP20OSV-ART; ISBN: 978-1-7281-5464-0

Study of Extractive Text Summarizer Using The Elmo Embedding

Hritvik Gupta
5th Sem B. Tech, Computer science and Engineering
Geetanjali Institute of technical studies
Udaipur, Rajasthan
hritvik7654@gmail.com

Mayank Patel
Associate Professor, Dept. Computer science and Engineering
Geetanjali Institute of technical studies
Udaipur, Rajasthan
Mayank999_udaipur@yahoo.com

***Abstract*— In recent times, data excessiveness has become a major problem in the field of education, news, blogs, social media, etc. Due to an increase in such a vast amount of text data, it became challenging for a human to extract only the valuable amount of data in a concise form. In other words, summarizing the text, enables human to retrieves the relevant and useful texts, Text summarizing is extracting the data from the document and generating the short or concise text of the document. One of the major approaches that are used widely is Automatic Text summarizer. Automatic text summarizer analyzes the large textual data and summarizes it into the short summaries containing valuable information of the data. Automatic text summarizer further divided into two types 1) Extractive text summarizer, 2) Abstractive Text summarizer. In this article, the extractive text summarizer approach is being looked for. Extractive text summarization is the approach in which model generates the concise summary of the text by picking up the most relevant sentences from the text document. This paper focuses on retrieving the valuable amount of data using the Elmo embedding in Extractive text summarization .Elmo embedding is a contextual embedding that had been used previously by many researchers in abstractive text summarization techniques, but this paper focus on using it in extractive text summarizer.**

Keywords*— *Text summarization, Extractive text summarization, Natural language processing, cosine similarity, Elmo embedding

I. INTRODUCTION

Over the last few years, information on the internet is growing by the tremendous rate with the growth in online education services, social media and all the professional fieldwork. This information is growing mainly in the form of the textual data. And so handling and understanding the one such large data has become the major issue. To overcome one such problem the one solution is to reduce the size of the textual data, in other words converting the long textual data into the summaries, which commonly called summarization[2].However, simplification of data into the short summaries is not that easy. It requires the proper understanding of the text document that are going to be summarized. This problem can be overcome by the Using the one of the approach Natural language processing that is text summarization discussed later in this section

Natural language processing (NLP) [1] is a field of computer science and artificial intelligence. The ultimate objective of the NLP is to read and make sense of the human language. The human language is commonly termed as natural language. Natural language processing has also its sub-branches 1) Natural language Understanding, 2)Natural language generation. A linguistic concept like part of speech structure and grammatical structure is commonly used in NLP. NLP has emerged and evolved in various phase over the last few decades. The version of NLP that is being used today is a deep neural network style and representation learning as it was able to achieved the state of art result in many NLP tasks.

Text summarization[4][7] is one of the major tasks of Natural language processing. Research In the field of the text summarization has been under investigation using different methods and algorithm over the last few decades. Text summarization is converting the long pieces of text into a coherent and fluent summary. In other words, extracting of only valuable information from the text document. The text summary that is generated conveys the most important information of the text document. The main task is to generate a summary that has 2 main features. Firstly the summary generated conveys some meaning and have only relevant information and secondly, the summary should be short in size. Text summarization is further divided[7] into the 2 subtypes Extractive text summarization and Abstractive text summarization. Abstractive text summarization[12][13] is done by using the semantic representation of the sentences to generate the summary that is closet to what human might create. In other words, abstractive text summarization generates the summary on its own. The word sequences might or might not present in the original text. Whereas Extractive text[11][10] summarization consist of selecting the only relevant information from text like sentences, words and phrases. It mainly comprises of the extraction of the sentences.

On the other hand, Contextual embedding[8][6] that is being used in this paper is the term that is most commonly used by the NLP researchers. Contextual Embedding is the assigning of the each word a representation (vector) based on its context, thereby capturing use of words across varied contexts and encoding knowledge that transfer across languages. In other embedding like Word2Vec , Glove embedding the words are encoded without capturing the knowledge of its environment. Same words might have different meaning in different sentences, Therefore for that reason, the contextual embedding has been used in this paper for the word vector representation.

The very known and Popular contextual embedding is the ELMO embedding that is most used by the NLP researchers today. Allen AI research team published a paper named "Deep contextualized word representations" [6], where they introduced a new type of deep contextualized word representation that models both the complex characteristics of the word used like syntax and semantics, and also how these uses vary across linguistic contexts (i.e., to model polysemy), and this new representation is called Elmo (Embedding from language models).

In this paper, the experiment for extractive summarization has been demonstrated using the Elmo embedding and scoring the sentences using Cosine similarity[3]. Although the most important task in any NLP algorithm or method is Text Pre-Processing. The illustrate fig.1 in section 3 describes the preprocessing of text. For better Extractive text summarizer Kulkarni and Apte in 2013 [4], provides 4 steps which include 1)Text preprocessing 2)feature extraction of sentences 3) sentences selection 4) summary generation. These steps have its respective task . fig2. Demonstrate the flow of the model in section 3

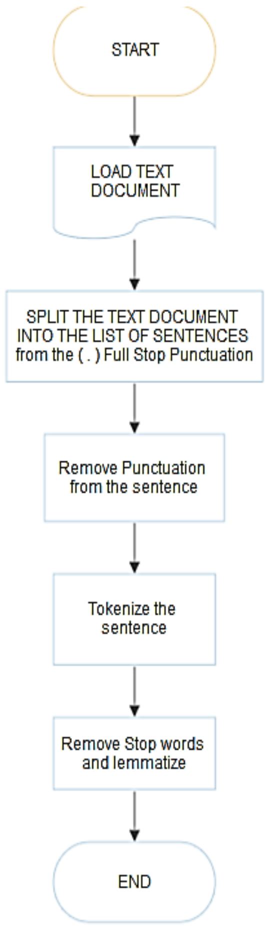
II. RELATED WORK

There had been much previous research work on automatic text summarization using natural language processing. Text Rank algorithm [5] and text summarization using the k-means clustering[14] is the automatic extractive text summarization techniques that use the inverse document frequency vectorizer technique(TFIDF)[9] or Count vectorizer to encode the text document for further scoring and ranking of text sentences in Single text Document. That was the basic approach which was followed by many NLP researcher for text data encoding in extractive summarizer approach. Using this vectorizer approach to encode the text data and then scoring and generating summary without knowing the word environment will lead to ambiguous summaries. For that reason, the contextual Elmo embedding has been used for text data encoding in extractive text summarizer.

Whereas, the word Embedding means representing the word vector in space to capture its linguistic properties[13]. Contextual Elmo Embedding[6] That is discussed later in the next section had been used previously in abstractive text summarization[8] technique. Which uses the pointer generation approach to generate the summary more preciously than other models. This approach uses the RNN for text generation[19]. But in this paper, the contextual embedding technique is used in contrast to extractive text summarization.

III. METHODOLOGY

Various algorithm[7] that had been used previously for automatic extractive text summarization. As discussed in the previous section, which is used to judge the text document by the size of the text and word that it contains. The researchers earlier worked on the extractive text summarization worked on a simple informative feature of the text frequency of words altogether they contain, or key phrases that specified the significance of the sentence.
This experiment aims to help the users to retrieve the valuable information and efficiently read the documents through summaries created by the model. The summary must convey the relevant meaning of the document. Our proposed method uses the Elmo embedding for converting the raw texts to the vector. The Elmo embedding[6] model is used to make the data more accurate as because it uses the bidirectional LSTM which capture the context-dependent aspects of the word meaning and aspects of the syntax of the words. So in short Elmo used to represent the token using all these peace of information in the single vector. Elmo Embedded vector obtained is then scored using the cosine similarity function. Each sentence will be scored against other sentence. Once the score of the sentences is receieved, then the sentences are ranked by sorting them in decreasing order and finally, the top 5 sentences with highest score is going to be picked up, and combine them to form the summary.
This paper divides the whole process into the four steps 1) preprocessing the text 2) feature extraction using the Elmo embedding 3) finding the score of sentences cosine similarity of embedded vectors 4) sorting the score and combining the top sentences to generate summary



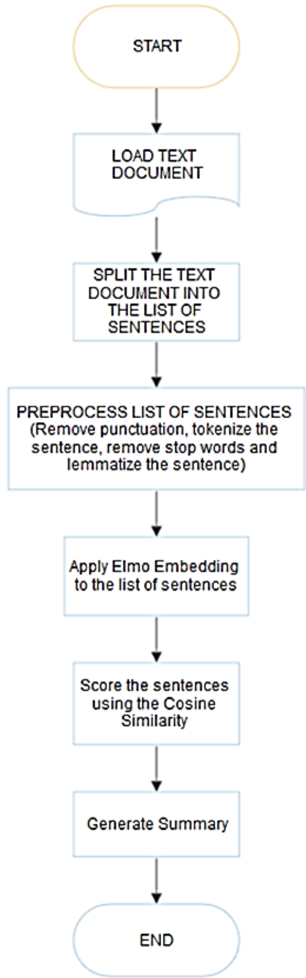


Fig1Preprocessing the text FIG.2 Flow of the proposed algorithm.

A. PREPROCESSING THE TEXT

Preprocessing[16] is one of the major tasks in the NLP for getting better and accurate results. It helps the algorithm to learn weights accurately. In our model, the document to be summarized contains n number of sentences and Elmo Embedding takes the list of sentences to convert it into the vectors. So the first task of the preprocessing is to split the text document into the list of sentences. There are many ways to splitting into sentences like using NLTK sentence tokenization, But our method gets more accuracy when the sentences are split from (.) full stop punctuation. After splitting, the next step is the process the document through NLTK. The processing includes the 4 steps which is applied to each sentences from list of sentences

1) *Remove Punctuation*: Take each sentence from the list of sentences, and lower case each word from the sentence and remove punctuation. Removing punctuation is one of the major tasks as it helps the model to vectorize the data more accurately. Punctuation removal involves the removal of '?', '/', like character

2) *Tokenizing the sentence*: Tokenization is the most common term that came across in natural language processing this simply means dividing the sentences into the list of words for further preprocessing. Therefore for the purpose of removal of stop words and Lemmatization, each sentence of document is tokenized one by one, then apply next steps.

3) *Stop words removal*: After tokenizing the sentence another task is removing the stop words which includes the 'the', 'if', etc. Stop words removal allows the model to capture the important words and phrases. Also, these stop words are one of the main reason for the generation of ambiguous summary.

4) *Lemmatizing*: This task includes the process of grouping together the inflected forms of a word so they can be analyzed as a single item. And aims to remove the inflectional ending only and returning the base. Finally combining the tokens to form the sentence. Therefore to short the long peace of sentence and for better data encoding, lemmatizing is used.

B. CALCULATE THE ELMO EMBEDDING

Elmo Embedding used to aggregate all the information of the token in the single vector. For that purpose, the Elmo model uses the Deep BILSTM[6] layer method where higher layer captures the contexts dependent aspects of the token and lower layer captures the aspects of the syntax

From the preprocessed list of sentences, Elmo embedding can be calculated. Elmo embedding model takes the list of sentences to return the array containing the vector representation for each sentence and shape of an array is (length of the list , 1024). More generally, to use Elmo for a specific downstream task, word representations are computed by a weighted sum of each intermediate network representation as shown :

$$ELMo_k^{task} = E(R_k; \theta^{task}) = \gamma^{task} \sum_{J=0}^{L} s_j^{task} h_{kj}^{LM} \quad \text{-(1)}$$

Here, s_j^{task} are Softmax-normalized weights, γ^{task} is the scaler parameter. γ is of practical importance to aid the optimization process. This eq(1) is the learned weighted average of all the representation.. shown the paper "Deep Contextualized word representation".[6]

C .SCORING AND RANKING THE SENTENCES

Cosine similarity[3] is the measure of the similarity that can be used to compare the vectors of words, sentences or document. For example let A and B be 2 vectors for comparison, having using cosine function as a similarity -of the function

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

The measure in eq(2) compute the angle between A and B. A cosine value of 0 means both the vectors are 90 degrees to each other and have no match. But the closer the value to 1, the smaller the angle and greater the match.

Once the Elmo embedding has been got, the sentences can now be scored and ranked. Scoring of embedded vector is done by using COSINE_SIMILARITY on each sentence vector against all the remaining sentences vector. In short , embedded vectors of like sentence 1 is scored against all the remaining sentences vectors.

D. SORTING AND SUMMARY GENERATION

The next step after scoring the sentences is to sort the scores. The sentences are sorted in the descending order. The sorting is important to analyze the rank of the value of the Elmo embedding of sentences. Finally, the top five sentences are chosen. As this method have been used is the extractive text summarization method, the sentences appear in the summary are same as the original document . These chosen final sentences are then combine to form the summary of the text.

IV. RESULTS AND DISCUSSION

This program is created using the python programming language and written in the Spyder console using the anaconda environment. The packages that are used is NLTK for text processing, Tensor-flow hub for downloading the Elmo pretrained model, cosine similarity from the Sklearn library and other common libraries like a panda for importing the dataset, Numpy etc. Once all the libraries are imported the next step is to import the data set, for the purpose of summarization the news summary data set have been used from the Kaggle datasets . The dataset contain headlines, complete text, summarized text , and news articles link for each article in the dataset.

In this experiment, the first 100 documents have been executed from the news summary dataset. And evaluate the result by using the Rouge scorer [16] method on the generated summary and the original summary. The statistical detail of any three document from the news summary dataset are represented are displayed with the Rouge-1 , Rouge-L, score including their precision, F-measure, recall in table 1 and table 2 . And the average Rouge score of first 100 , 50 and 30 documents are shown in table 3 and table 4.For the purpose of the comparison score of how much our purposed model is performing, having the Text rank algorithm [5].

1. *TABULAR RESULT REPRESENTATION*

Table-1 : (F = F-measure, P = Precision, R = Recall Rouge-1, and Rouge-L score of random 3 documents for extractive summarizer using Elmo)

DOCUMENT	***ROUGE-1***			***ROUGE-L***		
	F	***P***	***R***	***F***	***P***	***R***
Doc -1	0.60	0.48	0.81	0.41	0.35	0.52
Doc -2	0.46	0.37	0.61	0.37	0.31	0.46
Doc-3	0.53	0.43	0.72	0.53	0.43	0.67

Table-2 : ((F = F-measure, P = Precision, R = Recall) Rouge-1, and Rouge-L score of random 3 documents for Text rank algorithm

DOCUMENT	***ROUGE-1***			***ROUGE-L***		
	F	***P***	***R***	***F***	***P***	***R***
Doc -1	0.38	0.3	0.5	0.22	0.18	0.27
Doc -2	0.40	0.36	0.58	0.25	0.21	0.32
Doc-3	0.33	0.26	0.43	0.26	0.21	0.32

Table-3 (Average Rouge-1, Rouge-L score of first 100 documents, 50 documents, and 30 documents for Extractive text summarization using Elmo Embedding.)

NO. of Documents	***ROUGE-1***			***ROUGE-L***		
	F	***P***	***R***	***F***	***P***	***R***
100	0.44	0.36	0.56	0.37	0.31	0.46
50	0.45	0.37	0.59	0.39	0.33	0.48
30	0.46	0.37	0.60	0.39	0.32	0.49

NO. of Documents	***ROUGE-1***			***ROUGE-L***		
	F	***P***	***R***	***F***	***P***	***R***
100	0.38	0.31	0.49	0.29	0.29	0.36
50	0.38	0.31	0.49	0.29	0.25	0.37
30	0.39	0.31	0.49	0.29	0.24	0.36

Table-4 (Average Rouge-1, Rouge-L score of first 100 documents, 50 documents, and 30 documents. For text rank algorithm)

Rouge-L: measures the longest matching sequence of a word using the LCS. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence-level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length

Rouge-1: refers to the overlap of ***unigrams*** between the system summary and reference summary

Precision is helping us to tell how much the system generated summary is needed or relevant. Precision is measured as the number of overlapping words divided by the total number of the words In the system generated summary

$$\text{Precision} = \frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$$

Recall in the context of ROUGE means how much of the reference summary is the system summary recovering or capturing

$$\text{Recall} = \frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary}$$

F-measure is a relationship between the recall and precision which represent the accuracy of the system

$$\text{F-Measure} = \frac{2\,xPrecisionxRecall}{Precision+Recall}$$

In table 1 and Table 2, the Rouge-1 and Rouge -L score of any three documents have been calculated from the news summary dataset using both the algorithms, Extractive text summarization using Elmo embedding and Text rank Algorithm from paper[5], In table 1, observed that document 1 has highest rouge-1 score of all the three documents . This is due to the fact that there are large number of overlapping of unigram between system generated summary and the reference summary. Which in short shows Document 1 has higher F-measure or system generated summary has higher accuracy. And apart from the Document-1, it is observable that Rouge-1, and Rouge-L measure are of almost same for the other documents. Whereas, if Table 2 is observed, the score of the same document using the Text rank algorithm is low as compared to that of Extractive text summarization using Elmo embedding.

In table 3 and table 4, the average Rouge-1, and Rouge -L score for different sets of documents have been calculated for both the algorithm, In table 3, observed that in first 100 documents recall is higher than f-measure and precision, which means our algorithm is capturing/recovering of around

60 percent of summary from the reference summary. But on decreasing the set of documents to 50 and 30 the recall average increases slightly in percent. So it is observable that this algorithm is able to capture major part of relevant amount data from the text document. Whereas same trend is followed for the text rank algorithm as seen in table 4 , but here the score of the text rank algorithm is seen low as compared to Extractive Elmo model summarization algorithm . Which simply implies Extractive text summarization using Elmo embedding algorithm is performing well and same for all documents for capturing the relevant information

2..Graphical Result Representation

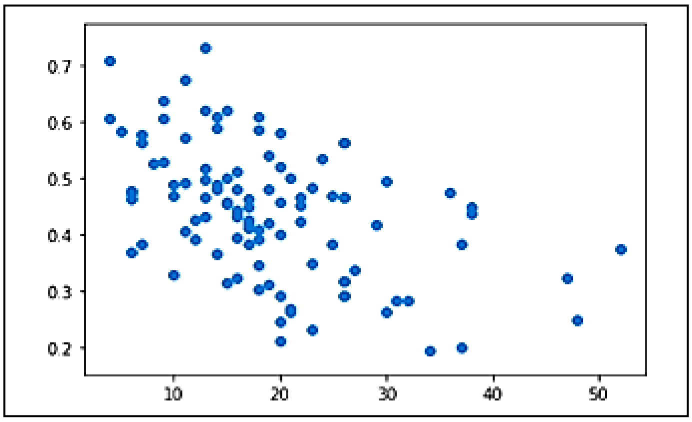


Fig3. (Elmo embedding model (F-measure rouge-1))

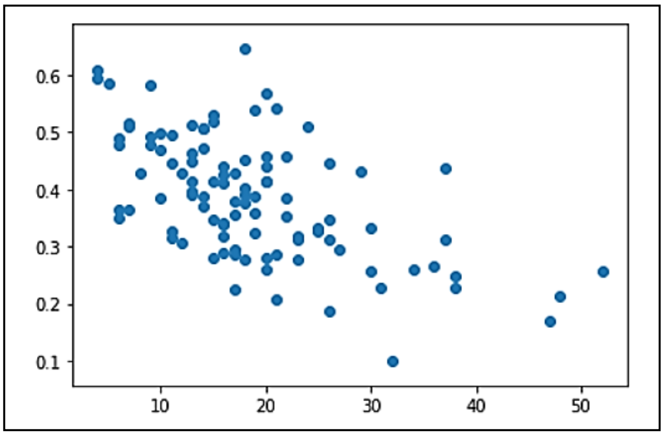


Fig4(Text rank algorithm model F-measure rouge-1)

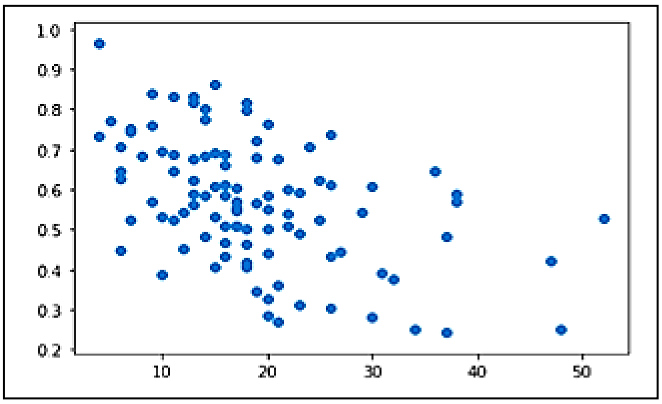


Fig 5 (Elmo embedding model Recall rouge-1)

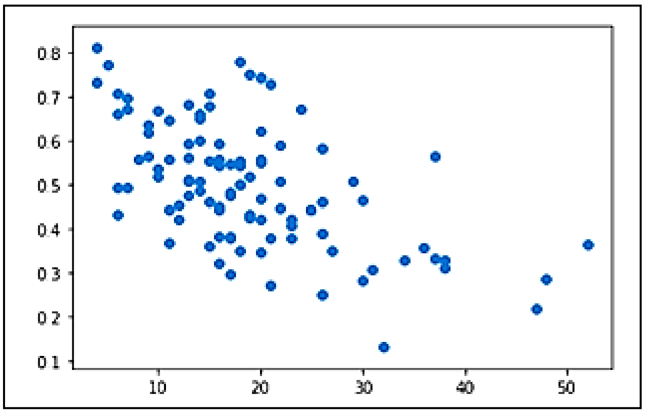


Fig 6 (Text rank embedding Recall rouge-1)

The Figures illustrated above are plot against the score and number of sentences of the first 100 documents, of the news summary dataset. A number of sentences of the text document are calculated after splitting the text document from (.)full stop punctuation. As described in the methodology splitting the text document gives the higher accuracy

Fig3 and Fig4 illustrate the F-measure score of Rouge-1 of first 100 documents of the news summary dataset vs the number of sentences in the document It can be seen that Extractive text summarization using the Elmo Embedding has the higher F- measure of a rouge-1 score as compared to the Text rank algorithm proposed in the paper [5].In other words, it means our proposed algorithm can achieve a higher range of accuracy.

Fig 5 and Fig 6 illustrate the Recall score of Rouge-1 of the text document of the first 100 documents vs the number of sentences in the document. It can again be observed that Elmo embedding extractive text summarization model performs well as compared to that of the text rank algorithm, as a recall for smaller length in Elmo is nearby 0.9 whereas, text rank algorithm has only reached 0.8. and the lowest score for text rank is 0.1 whereas, for Elmo, it is 0.25, Therefore it is observable that more number of major or relevant words are captured by using the Elmo Embedding Extractive text summarization

So it observable from the experiment conducted that:

1) Extractive text summarization using the Elmo embedding can get higher accuracy

2) Formation of a summary that is relevant to the user is done properly

3) Maximum information can be made available using less time.

4) Fast, effective and compact summary can be delivered using this automatic extractive text summarizer

5) Studied the new approach that Elmo Embedding that can not only be used in the abstractive text summarizer but also on Extractive text summarization

V. CONCLUSION AND FUTURE SCOPE

The research explains the use of the Elmo Embedding for Automatic Extractive text summarizer. Through this experiment, it can be seen that Elmo embedding can be used to encode the text document into the vector that contains all the necessary information of the word (contextual- dependent aspects and syntax) for the automatic extractive text summarizer. These vectors are further passed to the cosine similarity function so as to score and rank the sentences of the text document . Then the summary is able to be generated successfully by selecting top 5 sentences with the highest score. Elmo Embedding is Fast and reliable as it is able to capture the most of the relevant information or words from the text that are present in the summary, as seen from the recall which tells us that how much of the summary model is able to capture from the original summary. And also it can be seen that F-measure Using the Elmo Embedding has higher average score for the documents of larger length as well as for the shorter length as compared to Text rank algorithm[5] which uses the TFIDF vectorizer for encoding the data.

In simpler words, can be said that Elmo Embedding can not only be used in abstractive text summarizer for summary generations but also can be used it to extractive text summarizer for the purpose of the text ranking and summary generation. Elmo Embedding is proven to be convenient method that can be used to generate summary using Extractive method, But still there is a future scope of development of this Extractive summarizer, like creating the summarizer with much more accuracy and much faster way to process the data. Also, to improve this summarizer process multiple documents at once and give the generalized summary. The other improvement can be made like using the title of the text to summarize the text, As the title of the summary tells about the text like what the text about so using title it will increase the accuracy to summarize 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 S Jain1, 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] M R Prathima1*, H R Divakar2 Automatic Extractive Text Summarization Using K-Means Clustering in International journal of Computer sciences and Enginerring

[15] Vishal Gupta, G.Sl Lehal, "A Survey of Text Mining Techniques and Applications", Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1, pp- **60-76**, **August 2009**

[16] Chin-yew Lin, "A package for automatic evaluation of summaries",in Proc. ACL workshop on text summarization branches **OUT**,**2004**.

[17] Mayank Patel, Neelam Badi, Amit Sinhal (2019). The Role of Fuzzy Logic in Improving Accuracy of Phishing Detection System. International Journal of Innovative Technology and Exploring Engineering, **Volume-8 Issue-8**, ISSN: 2278-3075, pp. **3162-3164.**

[18] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK,pp- **347-352**, **2008.**

[19] hopra, S., Auli, M., Rush, A.M., 2016. Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. **93–98**

[20] Munot, N., & Govilkar, S. S. (2014). Comparative study of text summarization methods. International Journal of Computer Applications, **vol** 102(12), pp **33-37.**

[21] Karel Jezek and Josef Steinberger, "Automatic Text summarization", Vaclav Snasel (Ed.): Znalosti **2008**, pp.**1-12,** ISBN 978-80-227-2827-0, FIIT STU Brarislava, Ustav Informatiky a softveroveho inzinierstva, 2008

[22] Mayank Patel, Ruksar Sheikh (2019). Handwritten Digit Recognition using Different Dimensionality Reduction Techniques. International Journal of Recent Technology and Engineering. **Volume-8** Issue-2, ISSN: 2278-3075, pp. **999-1002**

[24] Logic. International Journal of Computer Engineering and Technology (IJCET), **VOL** 4(4), pp-**449-461**

[25] Data mining tools and challenges for current market trends, journel (IJSRNSC), **Vol 7** ,Issue 2 **,pp 11-14,** Apr-**2019**