Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)
IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7

METHOD OF TEXT SUMMARIZATION USING LSA AND SENTENCE BASED TOPIC MODELLING WITH BERT

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

2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) | 978-1-7281-9537-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICAIS50930.2021.9395976

Abstract— **Document summarization is one such task of the natural language processing which deals with the long textual data to make its concise and fluent summaries that contains all of document relevant information. The Branch of NLP that deals with it, is automatic text summarizer. Automatic text summarizer does the task of converting the long textual document into short fluent summaries. There are generally two ways of summarizing text using automatic text summarizer, first is using extractive text summarizer and another abstractive text summarizer. This paper has demonstrated an experiment in contrast with the extractive text summarizer for summarizing the text. On the other hand topic modelling is a NLP task that extracts the relevant topic from the textual document. One such method is Latent semantic Analysis (LSA) using truncated SVD which extracts all the relevant topics from the text. This paper has demonstrated 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. The algorithm proposed algorithm in this paper is able to achieve the score greater than that of text summarization using Latent Dirichlet Allocation (LDA) topic modelling.**

Keywords— Text summarization, Extractive text summarization, Natural language processing, cosine similarity, TFIDF Vectorizer , BERT, Truncated SVD

I. INTRODUCTION

In Past few years the population over the internet is growing by tremendous rate all over the globe with active 4.6 billion users. With such increase of the population over internet in the past few decades also results in the increase of the data, mainly text driven data. Fields like social media, news publishing platforms, and Google searches are some of the major platforms that are facing textual data overhead. Further these data are unstructured data that contains millions of relevant information that is needed to be managed and handled accurately. This is the point where natural language processing came into mainstream [1]. NLP helps in labelling and structuring highly unstructured textual data.

As due to increasing amount of text data generated every day, NLP became more important to make sense out of the data. The ability to make the data driven decision is crucial to any business, with every click on the internet data is generated and stored. To decrease the one such large amount of data over the data centers the field of the Natural language processing that is widely in use is Automatic text summarizer and Topic Modelling. Which reduced and size of data by retaining its relevant information and helps in structuring of the data.

Automatic text summarizer [2] is the sub-domain of Natural Language Processing, which is widely in use to make the shorter version of the long textual data. As automatic text summarizer decreases the size of data it also decreases reading time and increases the amount of relevant information in that area. There are mainly two approaches of automatic text summarization 1) Extractive text summarizer and 2) Abstractive text summarizer. Extractive text summarizer [3] involves extracting out relevant phrases, key words and sentences from the source textual document without making changes to them, further it combine them to make summary. On the other hand abstractive text summarizer is approach in which model is trained using deep learning algorithms which helps it create new phrases and keywords that relay on the information of the original text, and then it combines with some phrases or sentences from original documents to make up the summary. This research has demonstrated the experiment by using extractive text summarizer.

There are many ways in which extractive text summarization that can be performed in NLP, like using text rank method in which text summarization is done by capturing relevant sentences from the long text document by sentence embedding and scoring them using cosine similarity and then combine top sentences to form summary, other method is using topic modelling which captures the relevant sentences based on the topic which long textual document relay on. In this research we have demonstrated Extractive text summarizer using topic modelling and embedded the sentence using BERT large uncased.

Topic modelling [4] on the other hand is the task of NLP which automatically identifies the major topics in the long textual document. Topic modelling in NLP break down the text corpus to find semantic structure within the text and then to extract the topics the text corpus or document. There are various topic modelling algorithms [5] such as LSA using truncated SVD, LDA and etc. In this research paper we will be using former method LSA (Latent sematic analysis) using truncated SVD [6]. LSA is a model for extracting and representing the contextual meaning of words to compute the similarity between words, sentences or whole document. LSA uses TF-IDF for analyzing the text document and then learns topics by performing the matrix decomposition on document term matrix using singular value decomposition. LSA is used for reducing the dimension of the matrix to extract the topic from text document more precisely.

BERT (Bidirectional Encoder Representations from Transformers). It is the pre-trained deep learning model from unlabeled text. Developed and published by researchers at Google AI language [8]. BERT is a model that uses bidirectional layer with the attention model to language modelling. BERT uses the attention mechanism that learns contextual relation between the words in the text. BERT model is generally includes the contextual encoding of

sentences of long textual document. In this research work, we have uses the same contextual encoding mechanism that BERT performs for encoding the long textual document.

In this research we have demonstrated the experiment in which we are going to make the text summaries using Extractive text summarizer approach. The approach we are going to follow is to generate the position word embeddings using BERT of all the Topic word vectors of Document using LSA topic modelling and of Keywords Extracted using TFIDF vectorizer. And final scoring is made by comparing the positional embeddings using Cosine Similarity of Each sentence to that of the Topics extracted using LSA. Therefore in this way semantics of each sentences can be captured more accurately. All this is discussed in detail in next section.

II. Related Work.

Automatic text summarizer is a vast field for research in Natural language processing. There had been many previous research work that had contributed towards the development of automatic text summarization using topic modelling. Although Topic modelling came into existence in 2003, But gains a large amount of popularity in today's world, It aims to extract the topics from text document by understanding statistical relation among topics[7]. These topics can be further be used in Automatic text summarizer like Extractive text summarizer that are discussed in later sections.

Summarization using topic modelling aims to develop text summaries using 2 methods as discussed by Yihong Gong and ,Xon Liu in [10] first in which text summaries is generated by ranking the sentences and extracting relevant sentences from text document and second method is using LSA(Latent Semantic Analysis) to identify relevant sentences from the text document then combining them to form summaries. Another approach to text summarization using topic modelling is discussed in [11], in which text summarization is performed using term sentence matrix, in this topic embedded in the sentences of the document helps in sentence selection for automatic Extractive text summarizer approach.

Tong, Zohu &Zhang, Haiyi in 2016 published a paper [12] in which they use LDA (Latent Dirichlet allocation) topic modelling algorithm for extraction of set of keywords from a text document, then clustering the set of recurring keywords in group of sentences for text summarization. Although this model performed well in Wikipedia articles but still the major missing point is semantic structure of topics in each sentence of document The Bert model[8] developed by Google AI researchers, discussed the approach in which they uses bidirectional LSTM layer along with Masked LM (MLM) to capture the semantic and syntactic details of a text in the document .therefore BERT can be used as an encoder for creating the text embedding that contains the semantic and syntactic details of the text.

III. Methodology

The flow algorithm discussed in this research is represented in figure 1. The experiment is divided into 7 major stages. The main idea behind the algorithm is to generate the extractive text summary using LSA topic modelling[12] in an efficient way such that the generated summary contains all the relevant information about the text document, which we are able to achieve by combining BERT model for encoding sentences from textual document and TFIDF keyword extractor for extracting keywords from each sentences, to our text summarizer using topic modelling.

A. MODEL FORMATION

LSA topic modelling using Truncated SVD which is discussed in [13] used to extract the topic by generating the matrix that represent conceptually related sentences or documents and corresponding words. LSA assumes that these words will occur in similar pieces of text if they occur together and represent the same semantic meaning. SVD reduces this matrix for reducing dimension and noise, thereby capturing only relevant topics from the textual document. In this research sentences from a long textual document is passed to LSA topic modelling for finding the topic on which textual document based on.For a document D we name the number of sentences in it is (i) and these sentences is passed through LSA topic modelling.

TF-IDF for keyword extraction[15] is one of the major and one of the most unique part of this research. Used to extract the keywords from each (i) sentence of a text document . Each sentence can contains any number of keywords depending upon the length of the sentence. Number of keywords of each sentence (i) is represented by L

The purpose of the BERT encoder[7] 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. We will be storing these positional embedding for each topic and keyword extracted from LSA topic modelling and TFIDF keyword extractor respectively. These positional embedding are used for further calculation. Positional encoding of the word a vector of any word is represented by PE.

After Receiving the word embedding for each topic extracted using LSA Topic modelling from text document and for Keywords extracted from each sentences using TFIDF keyword extractor. 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 keywords from each sentences and store in Y(i), where i represent the sentence number for example Y(1) signifies the mean of word embeddings for each keyword in sentence 1 of text document.

Once value of X and Y(i) is determined the next task is to use the Cosine similarity[19] function to find compare the 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 combine to form required summary that contains the most relevant information about the long textual document.

B. TEXT PREPROCESSING

Text preprocessing[14] is the one of the major task of th Natural Language processing that perform of the task o

cleaning and removing the ambiguity from data for the efficient performance of an NLP algorithm. As algorithm learns weights more accurately when given right data. In our proposed model text preprocessing is performed by splitting the text document into the list of sentences and then passing these list of sentences into the text preprocessing. Text preprocessing involves 4 stages:-

1) Normalization: It is one of the basic step in any text cleaning task in which all the text in the data is normalized to same sequence of data, i.e. all data must be in same manner either all letter in the lower case or all letters in the upper case. For the betterment of the model generally all letters are normalized to the lower case.
2) Punctuation removal: For an algorithm to learn weight properly and for capturing semantic aspects of the text removing punctuation is one of the necessary task. Punctuation removal involves the removal of punctuations like '?', '!', '/' etc.
3) Stop words removal: stop words include words like 'a', 'an', 'in', 'is' etc. these are the useless data of any text document that doesn't contribute towards the semantic as well as syntactic aspects of the text document. These are removed by first tokenizing the sentence obtain by the punctuation removal then removing stop words by analyzing and removing each word that are present in the predefined list of stop words
4) Lemmatization: the last task of the text preprocessing is to lemmatizing the text. This is not the necessary step to apply but rather it generates the efficient output of the algorithm of the NLP algorithm. It involves morphological analysis of the words, and aims to remove the ending of the words that constitute same meaning.

C. EQUATIONS

1) Topic modelling using LSA : Latent semantic analysis permits the researcher to compare the semantic meaning of different parts of the text, by creating the term document matrix[16]. As the size of this textual document increases the dimensions of the LSA term matrix proportionally increases, therefore to extract the data containing relevant information or extract the topic from the term matrix Singular value decomposition(SVD) is used along with LSA[17]. The equation of reducing term document matrix 'A' , containing m sentences in rows and n unique words in columns into k number of topics using SVD is:

$$A = USV^T \quad (1)$$

The top k scored words for each m sentences is stored in value X.

2) TF-IDF keyword extractor : Term frequency inverse document frequency (eq 2) finds the TF-IDF score of the word w in a sentence i from list of m sentences in a document. For each sentence 'i' the TF-IDF score is used to find the relevant keywords for a sentence. Which then store in Y(i).

$$TF - IDF(w) = TF(W) \cdot IDF(w) \quad (2)$$

3) BERT encoder model for generating sentence embedding: The main reason behind generating the sentence embedding of each 'i' sentence in the text document is to capture the positional embedding of each topic and keyword extracted in previous steps. The positional embedding[18] (PE) of a word in the given sentence using BERT is given by :

$$PE(pos, 2i) = \sin\left(pos|1000^{2i/d}\right) \quad (3)$$
$$PE(pos, 2i+1) = \cos\left(pos|1000^{2i/d}\right) \quad (4)$$

This positional embedding are further used in next stages of calculations

4) Positional Embedding For LSA topics extracted:

$$sum = sum + PE([j]) \quad (5)$$
$$X = sum/k \quad (6)$$

- Equation 5 loop over all the extracted topics of text document and extract the positional embedding for each topic j, then it sum all the positional embeddings.

- Equation(6) take mean of all the positional embedding extracted by dividing sum with k number of topics extracted.

5) Positional Embedding for Keywords extracted:

$$sum = sum + PE([t]) \quad (7)$$
$$Y(i) = sum/L \quad (8)$$

- Equation (7) loop over every " i " sentence and find sum of Positional embeddings for every L number of keywords extracted using TFIDF keyword extractor.

- Equation (8) stores the mean value of position embedding for every keyword extracted in Y, where Y(i) represent mean values for each sentence " i " in the text document.

(6) Cosine Similarity :

Cosine similarity[19] is used as the scoring parameter which is used to find the difference between 2 vectors . In our case Positional Embedding is nothing but an array of vectors. In general Cosine Similarity is calculated as follow:

$$cos(\theta) = A.B\, cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (9)$$

$$Score(i) = Cosine\_similarity\left(X, Y(i)\right) \quad (10)$$

- Equation (10) loop for " i " number of times to compare the mean of Positional Embedding of k number of Topics Extracted which is X and L number of topics in each " i " sentences which is Y(i). These values are stored in Score(i).

7) Sentence Extraction and summary Generation:

The final step of the algorithm is to extract the sentences from "i" number of sentences of a text document for summary generation. That can be done by using the score(i) parameter calculated in previous step.

$$Index = \max(score(i)) \quad (11)$$

- Equation (11) is used to extract indexes of the sentences containing the maximum score. summary can be generate by combining the top 5 sentences containing the maximum score.

D. ALGORITHM

INPUT: Text Document D
OUTPUT: Summary
PROCEDURE:

a. Divide the Document D into the i number of sentence.
b. Pass these sentences to the text cleaning function.
c. Pass these sentences into the LSA topic Modelling Function for extracting k topics .
$$X = LSA_{model}(sentences)$$
d. Pass these clean sentences into the BERT encoder. For creating embedding:
$$Sent_{embds} = bert.encoder([sentences])$$
e. Pass clean sentence into TFIDF keyword extractor function for extracting keywords for each sentence i:-
$$Y(i) = tfidf_{keyword}\ Extractor(sentences)$$
f. $sum = 0$
g. $for\ every\ value\ of\ Y(i)$
$$L\text{-}Length(Y(i))$$
$$do\ sum = sum + Y(i)$$
$$End$$
$$Y(i) = sum/L$$
h. Initialize sum = 0
i. $do\ sum = sum + Sent_{embds}.encode([k])$ for every k word in X
j. Compute score using the cosine similarity function by looping every value of Y(i) and comparing each value with X
$$score(i) = 0$$
$$for\ every\ value\ of\ Y(i)$$
$$score(i) = Cosine_{similarity}(X, Y(i))$$
k. Set the score in descending order and extract top 5 sentences index with highest score.
$$index = \text{set}(\max(score(i)))$$
l. Combine the top 5 sentences from index and form the summary.
m. END

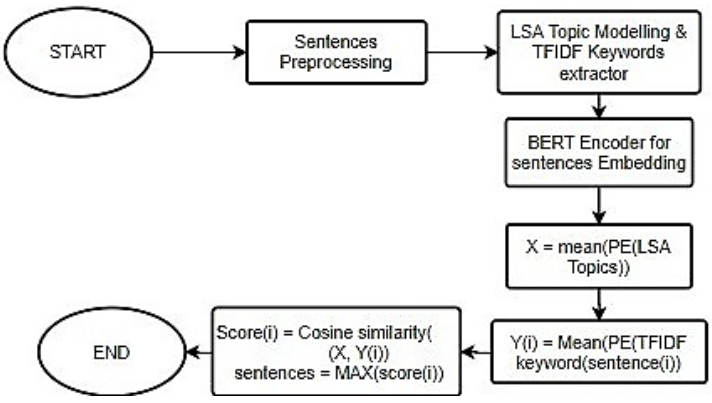

Fig 1. Flow of proposed algorithm

IV. Result and discussion

The experiment is conducted to figure out the best possible way to solve the extractive text summarization using topic modelling. As for purpose of evaluation we have selected the news dataset for text summarization from Kaggle dataset. The experiment is conducted over different sets of document. First experiment is conducted on 30 documents , then 50 and then 100 documents. Their cumulative score is calculated using the rouge evaluation [20] method Where ROUGE stands for the Recall-oriented understudy for Gisting Evaluation. A standard rouge package includes several measures to quantitively compare machine generated summary and the original summary. The evaluating parameters for this experiment is ROUGE 1 and ROUGE L. Where, ROUGE-1 measure the overlapping unigrams, in other words means it measure overlapping words between original summary and human generated summary. ROUGE-L measure the longest common sequence between the original and human generated summary. The major advantage of the ROUGE-L is that. there is no need to define the unigram length, it automatically selects the longest common subsequence. For the purpose of the comparison for the performance of the model we have taken another model Text summarization using LDA(latent Dirichlet Architecture) Topic modelling[21].The both parameter described gives the output in terms of the F-measure , precision and recall. Precision measure how much summary generated by the machine is relevant and useful.

$$\text{Precision(P)} = \frac{N_{overlapping}words}{Total_{summary}words}$$

Where, $N_{overlapping}words$ is total number of the overlapping words and Total words in machine generated summary.

Recall measure amount of relevant words captured by machine to generate the summary. And is given by:-

$$\text{Recall(R)} = \frac{N_{overlapping}words}{Total_{reference}words}$$

Where Total reference words is the total number of words in reference summary or in original summary. F-measure is the cumulative score of precision and recall. And is given by:

$$\text{F-measure} = \frac{2 \times P \times R}{P + R}$$

A. Tabular Representation Of The Result

Table-1 (Rouge-1, Rouge-L score of Random 3 documents from the dataset using proposed algorithm, F-measure, P- precision, R-recall)

DOCUMENT	ROUGE-1			ROUGE-L		
	F	P	R	F	P	R
Doc -1	0.70	0.56	0.90	0.69	0.57	0.89
Doc -2	0.65	0.51	0.89	0.64	0.54	0.83
Doc-3	0.66	0.53	0.88	0.68	0.56	0.86

Table-2 (Rouge-1, Rouge-L score of Random 3 documents from the dataset using text summarization using LDA topic modelling, F-measure, P- precision, R-recall)

Table-3 (Average Rouge-1, Rouge-L score of first 100 documents, 50 documents, and 30 documents using proposed algorithm)

DOCUMENT	ROUGE-1			ROUGE-L		
	F	P	R	F	P	R
100	0.43	0.35	0.58	0.36	0.30	0.46
50	0.45	0.35	0.59	0.37	0.31	0.47
30	0.46	0.36	0.58	0.38	0.32	0.48

Table-4 (Average Rouge-1, Rouge-L score of first 100 documents, 50 documents, and 30 document of text summarization using LDA topic modelling algorithm)

DOCUMENT	ROUGE-1			ROUGE-L		
	F	P	R	F	P	R
100	0.37	0.30	0.51	0.28	0.23	0.36
50	0.37	0.29	0.50	0.27	0.23	0.35
30	0.38	0.30	0.52	0.28	0.23	0.36

In Table-1, ROUGE scores of summarization using proposed algorithm is represented of any 3 random document from the dataset. As it can be seen that Recall scores of the documents is higher than the expected score. This Recall score can be interpreted as summary that is generated by using proposed algorithm by the machine is able to capture and extract the relevant and vital information from the original text. Whereas, for the purpose of comparison we have used another model which is text summarization using LDA topic modelling, where Recall score as seen in Table -2 is lower than that of the score of the proposed algorithm. Further this Recall score results in achieving higher F-measure. Therefore, our model is performing well in every aspect

In Table-3 and Table-4 we have depicted the cumulative ROUGE scores of different sets of documents from the dataset. Recall and F-measure score of the sets of document using proposed algorithm is higher as compared to that of the scores using LDA algorithm, Therefore it signifies that algorithm proposed in this paper are able to generate more number of overlapping words, or algorithm is able capture relevant amount of information from the document to generate summary. Here ROUGE-1 shows the number of unigram overlapping which is of same trend regardless of number of documents, this can be interpreted as algorithm behaving in similar manner regardless of the amount of information given to it.

B. Graphical Representation

In Fig 2, scatter graph for the Recall(ROUGE-1) score of first 100 documents against the number of sentences in each original text document to be summarized. This plot represents how accurately the proposed algorithm is able to retrieve sentences by extracting relevant and useful keywords present in each sentence. As seen in the graph for the text document

DOCUMENT	ROUGE-1			ROUGE-L		
	F	P	R	F	P	R
Doc -1	0.69	0.50	0.92	0.65	0.54	0.82
Doc -2	0.57	0.45	0.79	0.39	0.32	0.51
Doc-3	0.45	0.35	0.62	0.44	0.35	0.56

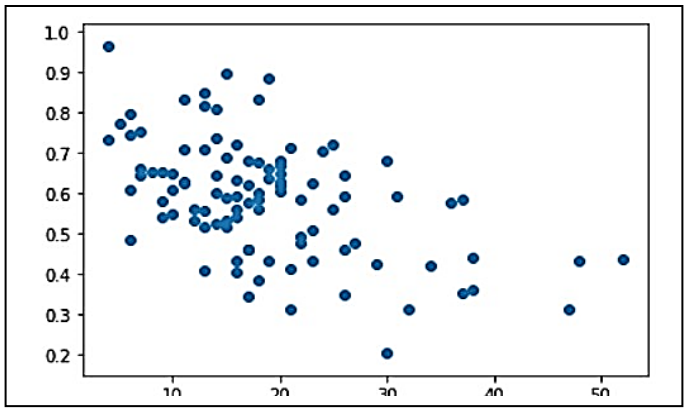

Fig2. (Proposed algorithm (Recall measure ROUGE-1))

containing lower number of sentences has much higher score than that of documents containing more sentences

Even though for document containing more number of sentences able to achieve appreciable amount of Recall Score which implies Algorithm is capturing good amount of useful or relevant to extract sentences from list of sentences to generate summary. It can be seen that as the number of sentences in the text document increases, the number of overlapping words decreases less as compared to number of words in reference summary, in other words number of overlapping words generated by algorithm is nearabout of same amount regardless of number of sentences as discussed in tabular representation.

In Fig 3, Graph represents the Recall of ROUGE-1 of first 100 documents against the number of sentences in the documents. This graph represents the score using Algorithm of text summarization using LDA topic modelling [2]. It can be seen that as the number of sentences increases the Recall score decreases with a higher rate. This implies that as the number of sentences in the text document increases the LDA algorithm could not been able to capture the relevant amount of sentences using topic modelling for text summarization. It can be seen that for the documents containing lower number of sentences the Recall score is similar that of the our proposed algorithm, but as the number of sentences increases, Recall score decreases with the a higher rate as compared to that of proposed algorithm. This means that the number of the relevant and useful words capture for extracting sentences to generate summary is far less.

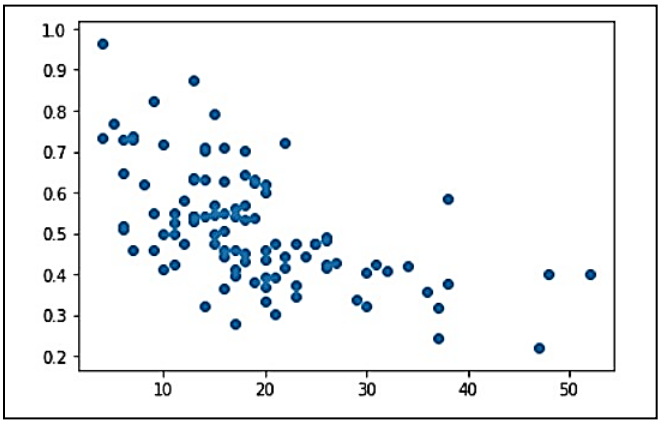

Fig3 (Text summarization using LDA topic Modelling (Recall measure ROUGE-1))

Therefore it can be observed from the experiment conducted that:

1) Using LSA topic modelling for whole text document along with sentence based topic modelling helps the machine to extract the sentences from the text document more accurately.
2) Using BERT for finding positional embedding for the topic word vectors extracted from whole text as well as from each sentence, results in finding accurate relationship between sentences and words.
3) From Recall score of Rouge-1 it can be seen that as the number of sentences in a text increases, number of overlapping words decreases less as compared to that of number of words in reference summary, therefore algorithm is performing accurately for large text.
4) The experiment conducted shows that maximum number of relevant information from the text document can be made available in concise summary.

V. Conclusion

This research explains the use of sentences keyword extractor and LSA topic modelling along with BERT on a text document results in extracting useful sentences from a text document that contains useful amount of information about the topic on which text document is based on. Recall score depicted that regardless of number of sentences of text document the relevant amount of information can still be extracted using proposed algorithm. The algorithm proposed were able to capture the semantic meaning of the topic word vectors in contrast with their semantic meaning to extract the sentences containing relevant number of sentences regarding those topics. LSA topic modelling using Truncated SVD when uses along TFIDF keyword extractor results in generating more accuracy for text summarization then LSA topic modelling alone. This tells us that each sentence contributes more to towards capturing semantic aspects of text than whole text alone.

The proposed algorithm results in achieving higher scores in different phases of experiment than that of when used LDA topic modelling for text summarization alone. This depicts that rather than creating clusters from topics received to extract the sentences containing semantic and syntactic aspects of text, we could generate the more accurate summary by comparing semantic aspect of each sentence with that of the topics from LSA model to generate the fluent and coherent summary. The future scope of this model is generating summary in more accurately, furthermore using proposed algorithm in abstractive text summarizer where machine is generating summary in its own language will might results in achieving greater accuracy.

VI. References

[1] A. Gelbukh, "Natural language processing," *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, Rio de Janeiro, Brazil, 2005, **pp. 1**

[2] A. P. Patil, S. Dalmia, S. Abu Ayub Ansari, T. Aul and V. Bhatnagar, "Automatic text summarizer," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, New Delhi, 2014, **pp. 1530-1534**

[3] Moratanch, N. & Gopalan, Chitrakala. (2017). A survey on extractive text summarization. **pp 1-6**

[4] Albalawi, Rania & Yeap, Tet & Benyoucef, Morad. (2020). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. Frontiers in Artificial Intelligence. **3. 10.3389**

[5] Rajasundari, T. & Palaniappan, Subathra & Kumar, Parambalath. (2017). Performance analysis of topic modeling algorithms for news articles. Journal of Advanced Research in Dynamical and Control Systems. 2017. **175-183**.

[6] B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, **2017**, **pp. 745-750**, doi: 10.1109/ICCONS.2017.8250563.

[7] K. Nokkaew and R. Kongkachandra, "Keyphrase Extraction as Topic Identification Using Term Frequency and Synonymous Term Grouping," *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Pattaya, Thailand, **2018**, **pp. 1-6**, doi: 10.1109/iSAI-NLP.2018.8693001.

[8] BERT : Pretraining of Deep Bidirectional Transformers for Language Understanding. Jacob devlin, Ming -wei chane, Kenton lee, Kristina Tautanova

[9] Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):**993–102**

[10] Gong Y, Liu X (2001) Generic text summarization using relevance measure and latent semantic analysis," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, **pp. 19–25**

[11] Wang, Dingding & Zhu, Shenghuo & Li, Tao & Gong, Yihong. (2009). Multi-Document Summarization using Sentence-based Topic Models.. ACL-IJCNLP. **297-300.**

[12] Tong, Zhou & Zhang, Haiyi. (2016). A Text Mining Research Based on LDA Topic Modelling. Computer Science & Information Technology. 6. 201-210. 10.5121/csit.2016.60616.

[13] Guo-Hua Wu and Yu-Tian Guo, "An enhanced LSA-based approach for update summarization," *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, **2015, pp. 493-497**

[14] Cherukuri, Aswani Kumar & Srinivas, S.. (2006). Latent semantic indexing using eigenvalue analysis for efficient information retrieval. Int. J. Appl. Math. Comput. Sci. **16. 551-558**.

[15] Li, Juanzi & Fan, Qi'na & Zhang, Kuo. (2007). Keyword extraction based on tf/idf for Chinese news document. Wuhan University Journal of Natural Sciences. 12. 917-921. 10.1007/s11859-007-0038-4

[16] Alghamdi, Rubayyi & Alfalqi, Khalid. (2015). A Survey of Topic Modeling in Text Mining. International Journal of Advanced Computer Science and Applications. **6**. 10.14569/IJACSA.2015.060121.

[17] Foltz, Peter. (1996). Latent Semantic Analysis for Text-Based Research. Behavior Research Methods. 28. **197-202.** 10.3758/BF03204765.

[18] 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**

[19] 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.**

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

[21] Rani, Ruby,Lobiyal, D. K.2019,topic-An extractive text summarization approach using tagged-LDA based topic modeling,jounnel-Multimedia Tools and Applications, 1573-7721,10.1007/s11042-020-09549-3 Rani2020

[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**