Impact Factor Value 4.046 e-ISSN: 2456-3463
International Journal of Innovations in Engineering and Science, Vol. 4, No.6, 2019
www.ijies.net

Comparative Analysis of Text Summarization Using Graph Based, Non-Graph Based and Deep Learning Methods

Shubham Kumar Pandey[1], Shital A. Raut[2]
[1]*M.Tech Student, CSE*
VNIT, Nagpur, India, 440010

[2]*Assistant Professor*
VNIT, Nagpur, India, 440010

Abstract – *With the rapid increasing text information on the web in the form of news articles, research papers, reviews, etc. the need for text summarization is felt. Text Summarization is the process of shortening the large document into a smaller one while preserving critical information. Although there are many approaches to summarize a document it's important to compare them in different aspects to conclude how the methods differ while summarizing a given document in terms of evaluation metrics, length of the document, etc. In this paper, we implement three different approaches Graph Based, Non-Graph Based and Deep Learning Method for summarizing a document and finally draw a comparison between their performance and also suggests the best model for summarizing a document in different use cases.*

Keywords- Text Summarization, Graph Based, Non-Graph Based, Deep Learning.

I- INTRODUCTION

The huge growth in information on the web in the form of news articles, research papers, e-books, emails, reviews, it becomes a challenging task for the human being to find out a piece of specific information. Because of such tasks, it creates the need for a system which can find out the specific and important information out of a large document. The solution to this problem is Automatic Text Summarization which is a kind of Text Mining task which mines the important information from a document either in the form of a complete sentence or by forming a new sentence.

Text summarization has many applications nowadays. It is used in providing small news headlines by summarizing entire news articles. In an E-Commerce website, text summarization is used for summarizing reviews of products. Query-based summarizer is used for finding out a specific piece of information from a document related to some query. Day by day, the application of Text Summarization is spreading into almost every field for minimizing the amount of information and to remove redundancy.

There are two ways to summarize the document one is the Extractive and another is Abstractive way. In Extractive way, the main focus is to extract the sentence for summary by finding out their importance using Statistical[1], Graph Based[2], Machine Learning or Hybrid Approach[3], etc, whereas in an abstractive way, the main aim is to form a complete new sentence of shorter length by finding out important information from multiple sentences using WordNet[4], Babel Net[5], etc, and finally merging them. Although Abstractive way creates a summary which is close to the human-generated summary but very few works have been done in this field because of the complexity and using highly advanced Linguistic features is not easy.

This work is a comparative study of 3 different models which are the extractive way of summarizing a document. Process of finding the importance of a sentence for a document differs for each model and it is also dependent on the length of the document. In this paper, we compared the performance of the models against each other and also compared their results with the length of the document. This work uses the Legal Database of Australia which

consists of Court Decisions of major court in Australia (AustLII). Each document also contains manually written catchphrases which will be used for evaluation of these models. The Paper is organized as follows: Section 2 describes the related work done in this field. Section 3 describes the methodology for comparative study. Section 4 suggests the evaluation method for the model. Section 5 discusses the result and section 6 concludes the study and suggest directions for future work.

II- LITERATURE SURVEY

Several works have been done in the field of Text summarization using different summarization techniques like Simple Statistics, Linguistic, Machine Learning or Hybrid approach. Extractive Summarization Techniques summarize the document by assigning a weight to important regions like Words, Sentences or Paragraphs of the document. And, finally sorting them according to their weight.

Saif, Quandell, and Martin[6] proposed a multi-graph based approach where the focus was to pick the best sentences by identifying the relationship between various sentence within the document and then constructing the graph out of it. Sentences which shares its context with maximum sentences within the document will be picked for the summary. They resolved the problem of finding context similarity by identifying the number of common words between the sentences. More the number of common words between two sentences more will be the similarity. The Graph is constructed where the number of edges between two sentences(nodes) in the graph is equal to the number of common words in both sentences. The total number of edges is stored in the symmetric matrix then summing up all the values in a row gives a score to every sentence of the document. Then, sentences are sorted according to their score and the sentences having the highest score picked for the summary. Main idea of this method is to find out which sentence shares common word with most sentences of the document. This approach sometimes neglects the importance of smaller sentence and synonym of a particular word. As a dataset, they collected 60 text passages and achieved average recall and precision of 0.65 and 0.07 respectively.

Flippo, Paul, and Achim[7] proposed a rule-based concept with a knowledge base where the focus is to build a rule by learning through the document itself. The approach is based on incremental Knowledge Acquisition where the initially any random rule is defined and the final rule is being built with incremental refinements from scratch, using the sentences within the document. Every relevant sentence is then matched with the initial rule, if the rule fails to qualify those sentences as a relevant one then it will be changed accordingly and again it will be matched with another relevant sentence. Once the rule is matched with every relevant sentence present in the manually created summary then we have our final refined rule. One issue with such an approach is that it needs manually created summary of the document to learn from. They applied this approach on 2816 cases from AustLII database and achieved average recall and precision of 0.29 and 0.87 respectively.

Siya and Manisha[8] proposed a rule-based concept where sentences will be picked according to a certain rule. Those sentences are going to be in the final summary which matches best with the rule. Initially, the rule is defined based on the value of certain features beforehand. Some features for every sentence is calculated. Low and High values of each feature are calculated and the initial rule is defined. Every Sentence is then passed through rule and each feature of the sentence is mapped with each feature of the rule. If there is a match, 0 is output otherwise 1 then all the 1's will be counted which provides a score to sentence. Hence, sentences will be sorted according to score and sentences with the lowest score will be picked for the summary. They applied this approach to 15 news articles from DUC 2002 dataset and achieved average recall and precision of 0.42 and 0.78 respectively.

Ladda, Mohammed, and Laomi[9] suggested summarization techniques using fuzzy logic. They calculated nine features for every sentence and extracted features are used as input to the fuzzy inference system. Using Gaussian membership functions every sentence get a value between 0 and 1. The obtained value determines the degree of importance of the sentence in the final summary. They initially stated input and output membership functions and based on that they defined IF-THEN rules for extraction of sentence. Extraction of the sentence from dataset from different field will need different IF-THEN rules. Once the rule is defined for a particular dataset then it can't be used for a different kind of dataset. Same problem is also there in method proposed by Siya and Manisha[8]. They applied this approach on 8 documents from DUC2002 dataset and achieved average recall and precision of 0.46 and 0.49 respectively.

III- PROPOSED WORK

For generating a summary of a document automatically, our summarizer will take the document as input and process it. The document will be then passed through pre-processing steps where all the irrelevant words will be removed from the document. Now, each sentence of a document will be passed through feature extraction steps where all the sentences will be represented by a vector. Then, the summary will be generated using three different methodologies Graph-based, Non-Graph based and Deep Learning based. All these methods assign a final score to every sentence of the document according to which sentences will be picked for the summary.

3.1 FLOW CHART

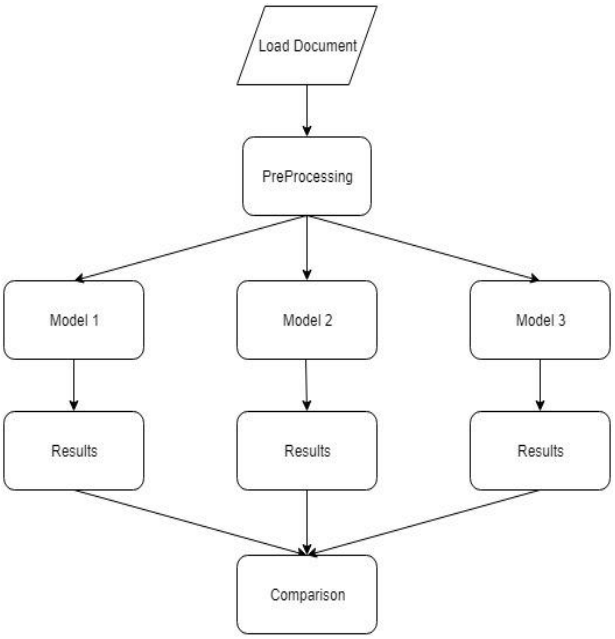

Figure 1: Flowchart of Methodologies

3.2 INPUT TO SUMMARIZER

Input to summarizer are Case reports from AustLII Dataset. Each report consists of 10 to 2900 sentences. Case Reports also contains manually generated catchphrases which will be used for evaluation purpose.

3.2 PRE-PROCESSING

Pre-Processing is the most important initial stage for summarizing the document using any summarization techniques. It is used to clean data or words from the document which is not required in determining the importance of a sentence. Input to this phase will be a list containing all sentences as a string and output will be a list of formatted Sentence. Applied Pre-Processing methods are listed below:

1. Part of Speech: POS Tagger is used to determine the part of speech of every word of a sentence and the word which is not either Verb, Noun, and Pronoun is removed from the sentence. *Word_tokenize* function is used to break the sentence into words and part of speech is being identified using a *pos_tag* function.

2. Stop word Removal: Sentence is tokenized into individual words and then commonly occurring words like a, an, the, as, etc, is being removed from the sentence. For this, all the stop words in the English language is downloaded using NLTK Library.

3.2 FEATURE SET

Each sentence of a document is represented by a vector where each value in a vector represents the score of individual features. For every sentence 12 features have been calculated. Input to this phase will be a list of formatted sentence and output will be a list containing feature vectors of each sentence.

1. **Proper Noun** – The number of proper Noun in a sentence. Usually, a sentence which contains more Nouns is an important one. The score of this feature is the ratio of the count of nouns in sentence divided by the number of words in the sentence.
2. **Has Cit Case** – Sentences which are citing to any other case or reports contain viable information about the document which can be inferred from the cited case or reports. So, such sentences are considered important. The score of this feature is 1 if a sentence is citing to any other case, otherwise, the value will be 0.
3. **Has hit Law** – Sentences which refers to an act or any section of the law are important because these sentences contain the reason behind filling a case and decisions of the court on that particular case. The score of this feature is 1 if a sentence is referring to any section of the law, otherwise, the value will be 0.
4. **Number of Title Word** – If a document has a title, generally the words in the title represent the main concept on which the document is based, so these words are important and are given extra weight. The score of this feature is the ratio of the number of words in the title

that occurs in a sentence divided by the length of the title.

5. **Total TF** – The number of times a particular word appears in the whole document will determine how important is that word to the document and summing up the count of all words in a sentence will determine the importance of a sentence to a document. The score of this feature will be Total TF of a sentence divided by maximum Total TF among all the sentences in the document.
6. **Average TF** - Sum of average term frequencies in the sentence.
7. **Total TFISF –** Term frequency indicates the frequency of a specific term within a document whereas this feature shows how frequently a keyword is used within all the sentences in a document. The score of this feature will be TFISF of a sentence divided by the maximum TFISF among all the sentences in the document.
8. **Average TFISF** - Average TFISF of terms in the segment.
9. **Total TLTF** – Sum (over all terms in the segment) of each term's frequency multiplies by its frequency in the segment. TLTF score of a word shows how important that word is to the sentence. The score of this feature will be Total TLTF of a sentence divided by maximum Total TLTF among all the sentences.
10. **Sentence Id –** Sentences that occur at the beginning and end are often the most important ones. So, the score of this feature will be 1 for the first and last sentence, 0 for the other sentences.
11. **Segment Length –** This feature is used to remove the small sentences from the document. The score of this feature will be the length of the sentence divided by the maximum length of sentence in the document.
12. **Significance Term –** Some specific words or phrases like 'court' or 'whether' increase the importance of the sentence. For this, all the legal terms have been taken and the score of this feature will be the count of all such legal terms in the sentence.

3.3 GRAPH BASED METHOD

3.3.1 Flowchart

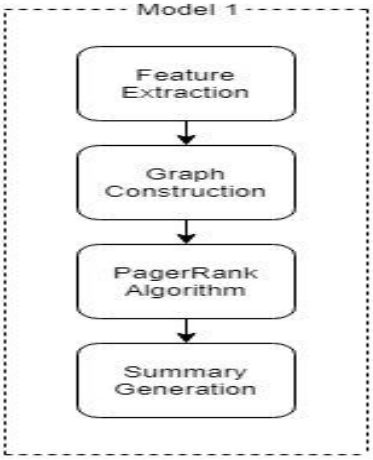

Figure 2: Flowchart of Graph based Model

3.3.2 Graph Construction

Graph based[2] ranking algorithm is essentially a way to rank vertex within a graph which in turns decide the importance of the vertex. To enable the application of the graph-based ranking algorithm to automatic text summarization, we have to build a graph that represents the sentence and inter-connectedness between them. For text-summarization, our aim is to rank sentence of a document and therefore a vertex is added to the graph for each sentence of the document.

Let G = (V, E) be a graph with a set of vertices V and a set of edges E where V = { $S_1, S_2, S_3 \dots \dots S_n$ } ; S_i is the i^{th} sentence. Every sentence is represented by feature vector $<f_{1i}, f_{2i}, f_{3i} \dots \dots f_{ni}>$. To establish affiliation between sentences of the document, 'similarity' relation is employed, wherever 'similarity' is measured as cosine similarity. The relation between two sentences may be seen as a method of advice to ask alternative sentences within the document that addresses the identical conception, and thus a link can be drawn between any two sentences that share common content. The degree of similarity between sentence S_i and S_j will become the weight of the edge between nodes representing the two sentences. If the similarity value is higher or equal to some particular threshold then there will be an edge. Input to this phase will be the feature vector of each Sentence and output will be Bi-directional weighted graph. For documents of AustLII dataset, keeping restrictions on similarity value for drawing an edge between two sentence removes an average of 67% edges from the graph.

$$\textit{Cosine Similarity}(S_i, S_j) = \frac{\sum_{b=1}^{n} f_{ib} * f_{jb}}{\sqrt{\sum_{b=1}^{n} f_{ib}{}^2} \sqrt{\sum_{b=1}^{n} f_{jb}{}^2}}$$

3.3.3 PageRank Algorithm

PageRank[2] is a mathematical formula based on a probability distribution that Google uses to calculate the importance of a specific page/URL. This formula algorithmic assigns every webpage a numeric value which is PageRank score of that Webpage. PageRank Algorithm can only be applied to the graph so, at first, graph is constructed where each vertex represents a Webpage. In our case, we will use bi-directional graph.

Initially, each node will get the initial value of 1 / S, where S is the Number of sentences in the document. After running the PageRank Algorithm in the document graph every node will have a final score. Input to this phase will be Bi-directional weighted graph and output will be a list of the final score for each Sentence.

$$PR\,(V_i) = c * \sum_{V_j \in in(V_i)} \frac{W_{ji} * PR(V_j)}{\sum_{V_k \in out(V_i)} W_{jk}}$$

3.3.4 Summary Generation

Once the final score for each Sentence is calculated, all the sentences will be sorted according to their scores. Based on the percentage of summarization those number of top sentences will be selected for summary and re-arranged according to their position in the document. Input to this phase will be the final score of each Sentence after applying PageRank Algorithm and output will be generated Summary for the document.

3.4 NON-GRAPH BASED METHOD

3.4.1 Flowchart

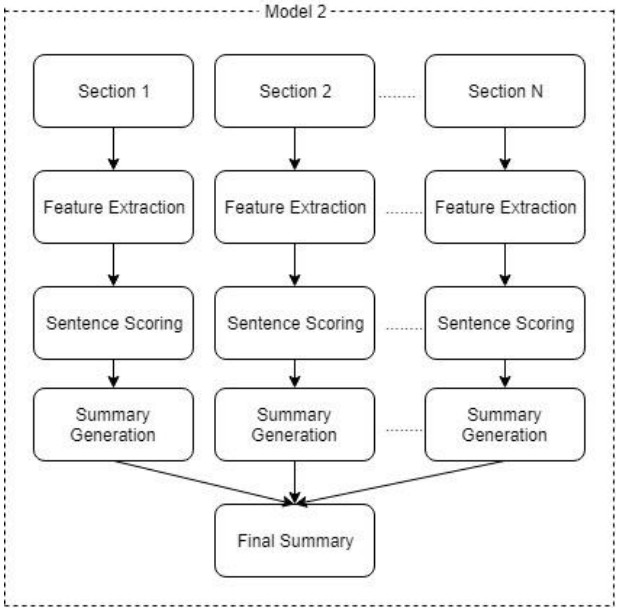

Figure 3: Flowchart of Non-Graph based Model

3.4.2 Multi-Document Summarization

Here we consider a single document summarization problem as a multi-document summarization problem [10]. In summary of any document, sentences have to be evenly distributed throughout the document in order to provide meaningful and sequential information. In order to extract sentences from each portion of the document, the document is divided into several smaller document depending on the length of the original document. It has the following steps:

1. It divides the entire document into a number of smaller sections depending on the count of the sentences in the document. It considers each smaller section as a separate document.
2. Then it repeats the feature calculations steps for each section.
3. Summary will be generated after sentence scoring phase where every sentence will be assigned a score based on the value of their feature.
4. Finally, it produces summary by combining the summary of individual sections.

3.4.3 Sentence Scoring

Once all the value of the features of a sentence is calculated in the previous steps. In this step, the score will be assigned to each sentence based on the linear function of its feature values. Every feature will be assigned a weight according to their importance. All the weights have been calculated practically. Total Score of a sentence is calculated by [11],[12]:

$$Score(S) = \sum_{i=1.....n} W_i * f_i\,(s)$$

where, W_i is the weight of i^{th} feature

3.4.4 Summary Generation

Once the score of each sentence of all the section is calculated then sentences of every section will be sorted according to their score and top sentences will be picked for the summary. Finally, summaries of all the sections will be combined to form a final summary.

3.5 DEEP LEARNING METHOD

3.5.1 Flowchart

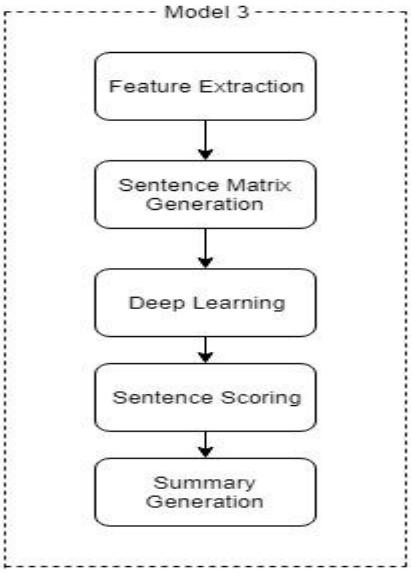

Figure 4: Flowchart of Deep Learning Model

3.5.2 Sentence Matrix Generation

After calculating the feature value of each sentence of the document then a two-dimensional matrix is generated where i^{th} row represents the vector having the value of all the features of the i^{th} sentence of the document. Sentence matrix, S = { $S_1, S_2, S_3 \ldots\ldots S_n$ } where S_i =[$f_1, f_2 \ldots\ldots\ldots f_{12}$] and n is the total number of sentences in the document. This sentence matrix will be used by the deep learning method.

3.5.3 RBM (Restricted Boltzmann Machine)

RBM [3] is an artificial neural network which consists of one visible layer and one hidden layer having the restriction that no two neurons of the same layer will be connected with each other thus, giving it a shape of a bipartite graph. In this case, RBM is used to enhance the values of the sentence matrix using learning methods in order to generate a more accurate summary. The neural network model is constructed where the number of visible layers is 12 corresponding to 12 features of the sentence and the number of hidden layers is 5 with a learning rate of 0.1. Input to this phase will be Sentence Matrix and output will be Enhanced Sentence Matrix.

3.5.4 Sentence Scoring

Once the final enhanced sentence matrix is generated by the RBM, the final score of each sentence is calculated. The score of i^{th} sentence will be the sum of all the values in the i^{th} row of enhanced sentence matrix.

Score $(S_i) = \sum_{j=1\ldots.12} w_{ij}$, *where* w_{ij} *is value at* i^{th}*row and* j^{th} *column.*

3.4.5 Summary Generation

Once scores of all the sentence is calculated they will be sorted according to the scores and the top sentences will be picked for the summary.

IV-EVALUATION

For evaluation of the summarizer, author-written Catchphrases will be used. Evaluation Method will be based on Rouge score which compares Catchphrases with the extracted sentences based on N Grams-statistics found to be highly correlated with human evaluations.

Every extracted Sentence will be compared with individual catchphrases and if the recall is higher than the particular threshold then it is considered as the match and that sentence is declared as a *relevant sentence*. Once every sentence is matched with each catchphrase, then the Precision of the summarizer will be the number of relevant Sentences divided by the total extracted Sentence whereas, the recall will be Matched Catchphrases upon total Catchphrases.

$$Precision = \frac{Relevant\ Sentences}{Total\ Extracted\ Sentences}$$

$$Recall = \frac{Matched\ Catchphrases}{Total\ Catchphrases}$$

V- RESULTS

This Approach is evaluated on 100 case reports from AustLII dataset. Each case reports, our summarizer generates summary with 10% compression rate. We used Rouge-1 with a similarity threshold of 0.5 to define a match.

Evaluation Metrics used for evaluation of the models are Recall, Precision and F-Measure. Recall, Precision and F-Measure values for 100 documents from AustLII dataset were calculated. Result of these three methods are now compared with the results of the method suggested by the Filippo, Paul and Achim [7]:

Table 1: Results of Graph based, Non-Graph based and Deep Learning Method.

	Recall	Precision	F-Measure
K.B	0.87	0.29	0.43
Graph based	0.56	0.51	0.53
Non-Graph based	0.64	0.70	0.68
RBM	0.59	0.58	0.58

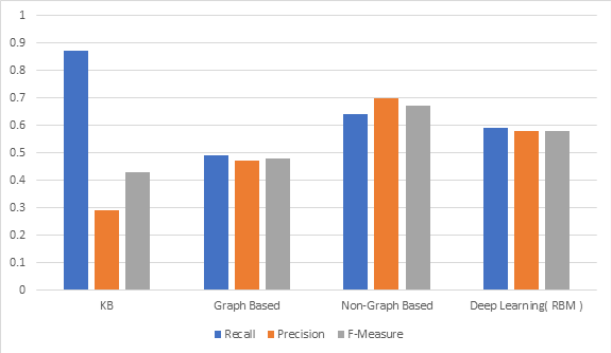

Figure 5: Comparison of Graph Based, Non-Graph Based and Deep Learning Method.

From the above results, it is clear that Non-Graph based method gives higher average recall, precision, and f-measure than Graph-based and Deep Learning Methods. The performance of these 3 methods varies according to the length of the document. So, in order to get a clear picture of their performances, we have compared their results with every range of document length. Below are the three graphs:

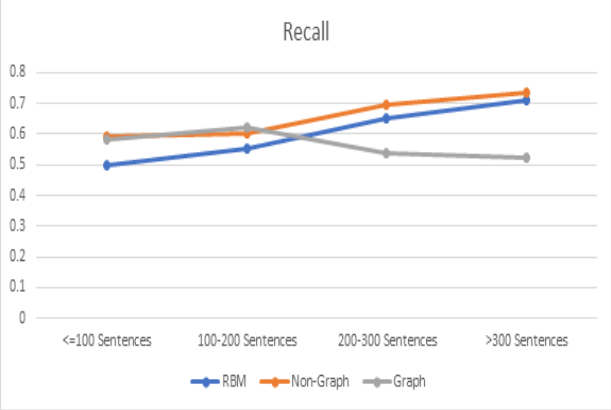

Figure 6:Recall of all models with different document length.

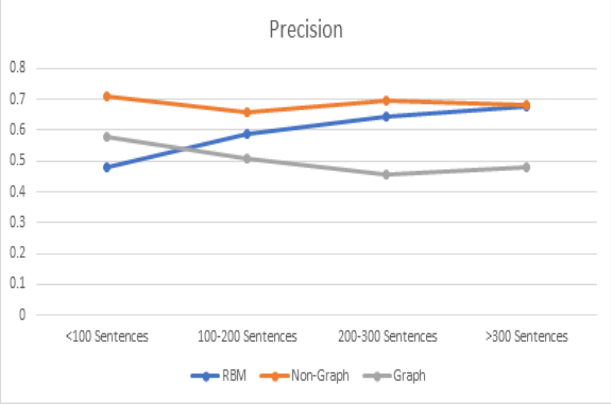

Figure 7:Precision of all models with different document length.

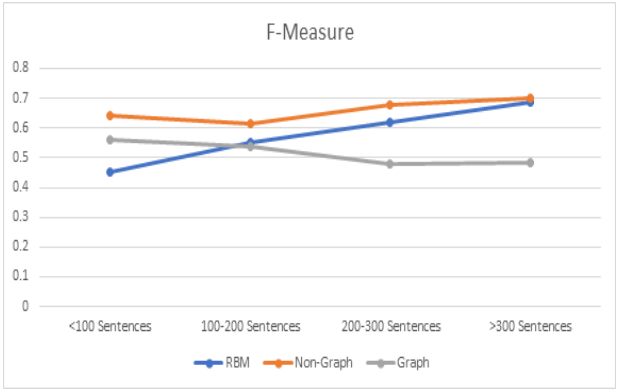

Figure 8:F-Measure of all models with different document length.

VI- CONCLUSION AND FUTURE SCOPE

From the above figure1,2,3, it is quite clear that Graph based algorithm performance decreases when the document size increases and for a smaller document it is better than Deep Learning Method. For medium sized documents, Non-Graph based method is better than the rest of the two. And, for the bigger document both the Deep Learning method and Non-Graph Based method perform well but the performance of Deep Learning method increases almost linearly with the increase in the document size while that's not the case with Non-Graph based approach. Therefore, with further increase in the length of document deep learning method will provide better results than Non-Graph based approach.

In this paper, we compared three different models for Text Summarization. All model uses the same feature set in which some features are restricted to be used only on Legal Databases. In the future, a diverse feature set can be selected which will be applicable to different kind of database consisting of news articles, sports articles, documentary, etc. And, then the comparison between these three models can be drawn to see which model gives the best performance for all kind of documents. There are many more approaches to summarize a document, in future some of the other models can be compared to get the best among all of them.

REFERENCES

[1] *Shohreh Rad Rahimi and Ali Toofanzadeh Mozhdehi, "An Overview of Text Summarization" 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI) IEEE 2017.*

[2] *Khusboo S. Thakar, R.V. Dharaskar and M.B. Chandak, "Graph-Based for Text Summarization" Third International Conference on Emerging Trends in Engineering and Technology IEEE 2010.*

[3] *Heena A. Chopade, Dr. Meena Narvekar , "Hybrid Auto Text Summarization Using Deep Neural Network and Fuzzy Logic System" Proceedings of the International Conference on Inventive Computing and Informatics IEEE 2017.*

[4] *Alok Ranjan Pal and Diganta Saha , "An Approach to Automatic Text Summarization using WordNet" International Advance Computing Conference(IACC) IEEE 2014.*

[5] *Haniyeh Rashidghalam, Mina Taherkhani and Fariborz Mahmoudi, "Text Summarization using Concept Graph and BabelNet Knowledge Base" Artificial Intelligence and Robotics IEEE 2016.*

[6] *Saif alZahir, Qandeel Fatima and Martin Cenek ," New Graph-based Text Summarization Method" Pacific Rim Conference on Communications, Computers and Signal Processing pp. 396-401 IEEE 2015.*

[7] *Filippo Galgani, Paul Compton and Achim Hoffman, "Combining Different Summarization Techniques on the Legal Text" Proceedings of the workshop on Innovative Hybrid Approaches to the processing of the Textual Data (Hybrid 2012).*

[8] *Siya Sadashiv Naik and Manisha Naik Gaonkar, "Extractive Text Summarization by Feature-based Sentence Extraction using Rule based Concept" 2nd International Conference on Recent Trends in Electronics, Information and Communications Technology pp.1364-1368 IEEE 2017.*

[9] *Ladda Suanmali, Mohammed Salem Binwahlan and Naomie Salim , "Sentence Features Fusion for Text Summarization Using Fuzzy Logic" Ninth International Conference on Hybrid Intelligent Systems pp. 142-146 IEEE 2009.*

[10] *P. Krishnaveni, Dr S.R Balasundaram, "Automatic Text Summarization by Local Scoring and Ranking for Improving Coherence" International Conference on computing Methodologies and Communication(ICCMC) pp. 59-64 IEEE 2017.*

[11] *Mahsa Afsharizadeh, Hossein Ebrahimpour-Komleh,Ayoub Bagheri "Query-oriented Text Summarization using Sentence Extraction Technique" 4th International Conference on Web Research(ICWR) 2018.*

[12] *Garcia-Hernandez, R.A and Y. Ledenva, "Word sequence models for single text summarization" in Advances in Computer-Human Interactions, 2009. ACHT09. Second International conferences on IEEE 2009.*