iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.46070

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Hybrid Semantic Text Summarization

Aesmitul Nisa[1], Mr. Ankur Gupta[2]
[1]*M. Tech Scholar,* [2]*Assistant Professor Department of Computer Science and Engineering, RIMT University, Mandi Gobindgarh, Punjab India*

Abstract: ***Automatic summarizing involves condensing a written material using a computer algorithm to provide a summary that keeps the key ideas from the original text. Finding a representative subset of the data that includes the details of the complete set is the basic goal of synthesis. There are two different sorts of summarising approaches: extractive and abstractive. Our system is interested in a mix of the two methods. To produce the extracted summary in our method, we have incorporated a variety of statistical and semantic variables. Emotions are significant in life since they reflect our mental condition. As a result, our syntactic characteristic is empathy. To creating summaries, our approach fundamentally integrates syntactical, psychological, and statistical techniques. We implement petroleum text summarization using word2vec (Deep Starting to learn) as a semantic feature, K-means clustering technique, and system parameters.***
The innovative speech synthesizer, which combines WordNet, Lesk engine, and POS, receives the created extracted analysis and converts it into an abstractive analysis to create a hybrid exhibited great.
Using the DUC 2007 dataset to assess our summarize, we produced effective results.
Keywords: Wordnet, Semantic, Test summarization, Abstractive

I. INTRODUCTION

In today's rapidly expanding current instance, document clustering has developed into a significant and useful tool for aiding and analysing text content. Currently, the traditional semantics challenge of data distillation from textual documents has received renewed focus as a consequence of the World Wide Web's exponential increase and accessibility to information. Fundamentally, this operation is a denoising procedure. Distilling the original text into a shorter version while maintaining its relevance and meaning is the aim of automated text processing. The method of manually constructing a portion of a text corpus while maintaining its contextual information is known as text summarizing.

Summarizing text automatically is a crucial topic of study in natural language understanding (NLP). Textual summarizing is on the rise and could offer an answer to the internet addiction issue. From a text, one may deduce many kinds of summarizes. Quick summary techniques include extracting, role in this type, and hybrid. In order to determine which statements are crucial for understanding a particular material in its whole, extraction summaries frequently rely on sentence separation processes. In order to create extractive summarization, important text portions (syllables or sequences) are taken apart from the text using empirical study of single or combined surface-level variables such descriptor abundance, placement, or cue words to identify the words that need to be taken out. Coming back to life the gathered content results in an abstractive summary. The bulk of the study is focused on extracting information from a given text using a few eye criteria, such as the placement of a phrase within the text, the formatting of terms (bold, italic, etc.), the incidence of a word within the text, etc. However, this method has a major flaw in that it heavily rely on the statement's formatting. Therefore, a phrase's relevance is determined by its structure and placement in the text as opposed to by its semantic content. We have concentrated on the schemas of utterances in the suggested strategy. Our technology is interested in a mix of both methods (Abstractive and Extractive). Using the concept of syllable ranking, we suggest an extractor method for image captioning. Statistics including sentence length, syllable position, periodicity (TF-IDF), group of words and verb phrase, and test set are used to rate sentences. In a bid to identify semantically significant phrases for the purpose of constructing a general extraction summary, we have additionally added a semantic feature using an unconstrained learning word2vec model and k-means clustering. This extractive summary is supplied to the basic language converter, which turns it into an input text synopsis using Wordnet and the Lesk procedure.

II. OBJECTIVES

The main objectives of this effort include :

1) To develop an extractor method for a successful image captioning.
2) To use extract summarization for empirical and original semantic data

3) To use sentence length, syllable position, periodicity etc as parameters for the rating of the sentences.
4) To use Word2vec learning model and k clustering for the language conversion.
5) To use the wordnet and lesk methods

III. LITERATURE REVIEW

When Luhn began his early work on automated text summary in the 1950s, sentence retrieval was born. He made reference to the use of individual words to determine which statements should be included in the summaries [1]. Words that feature prominently in the text are explanatory or theme words, and the sentences that combine these words are the prominent sentences.
Edmunson expanded on Luhn's study by pointing out that several characteristics might signify salient phrases. He has used following criteria to rank the utterances in a research source: (1) word severity, or the extent to which the word appears in the text; (2) statistic of title statements or header part words in the word; (3) comment placement in relation to the text and or the component; and (4) the number of drum roll, such as "in concluding, in recap" [2].

IV. METHODOLOGY

The work we've done combines both extractive and abstractive summarizing approaches. The most advantageous aspects of both strategies will be combined to create a system that is scalable, dependable, and more effective. We have largely focused on the mechanics of the text in our approach (i.e the meaning of the sentence). Because emotions are significant, they are given weight in sentences just like other factors. We may break our summarizing process into six components.:-

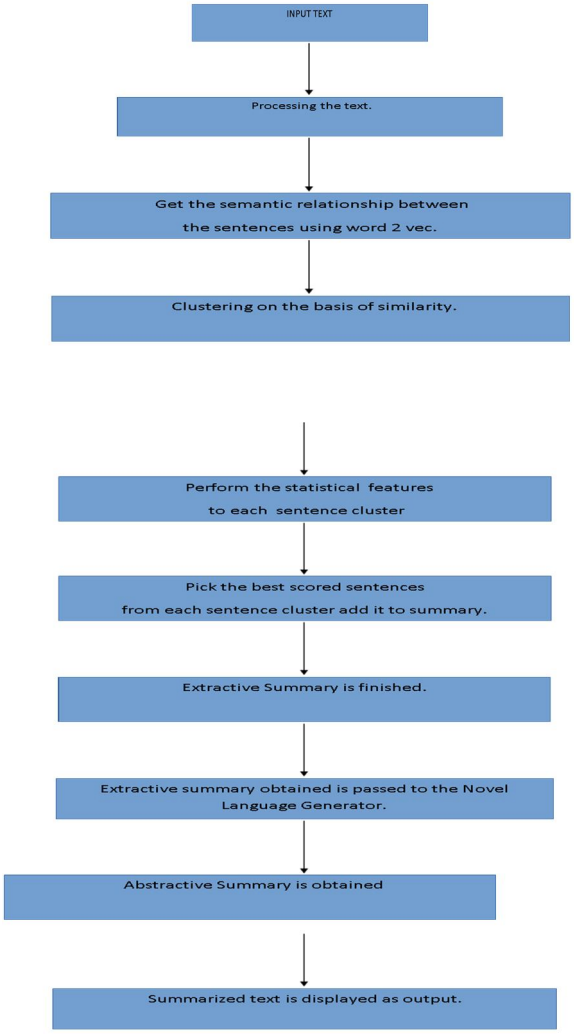


Figure 1 . Block diagram of the hybrid semantic text summarization

A. Pre-processing Of Text

The message that has to be summed up is first also before the or divided into sentences and then each phrase is broken down into words or tokens. The Java libraries and frameworks from Stanford Core NLP are used to complete it. Pre-processing is done on the text to get it ready for applying different approaches to get a hybrid summary.

B. Word Embedding

In this stage, word2vec is used to analyse the post data that was collected in the form of tokens in order to identify any semantic relationships. Word embedding is created using a collection of linked models called Word2vec. These models, which are two-layer shallower neural networks, have been taught to recover word contexts from linguistic data. Each distinct word in the corpus is given a corresponding vector in the space by word2vec, which receives its input from a sizable corpus of text (in our method, corpus DUC 2007 is utilized). In order to arrange all the words with comparable meanings in a single vector, word2vec groups the vector of related words together in vector space.

It works as follows:

1) Pre-processing is the first stage since Word2Vec has to be passed with words rather than complete phrases.

2) Using Word2Vec, comparable words for each token are determined.

3) • Create a statement using related words from each token and depict it using a large vector.

S1=T1,T2.....TN
S2=T1,T2.....TN
SN=TN1,TN+1.....TN
S1=T1=W1,W2....WN,TN=W1,W2....WNN V1.
SN=TNI=W1,W2...WN,TN=W1,W2,WNN VN

where S1, S2....Sn are sentences.

T1, T2...Tn are the tokens of each sentence. W1, W2...Wn are the similar words of each token.

V1,V2,....Vn are the big vectors of sentences.

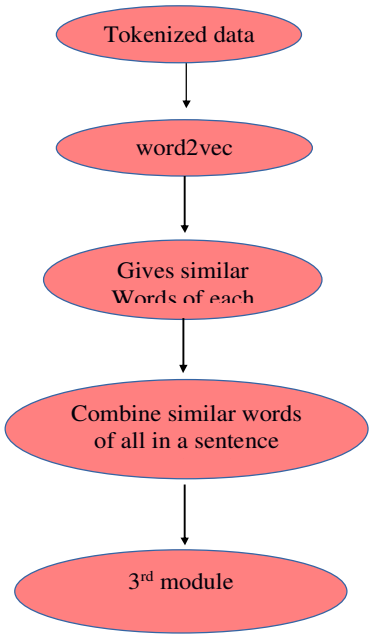


Figure 2 Representation of second module that is the word embedding

K-means aggregating technique is employed in our strategy. It is a method for unlabelled data that partitions the input into several categories (assume k cluster). In this method, related sentences are grouped together. We must create the vectors before clumping. This is accomplished by first computing the vectors' TF-IDF and then using the k-means method.

C. Clustering

The phrases with comparable concepts are grouped together to form the bunches created by the K-means automated system. The bunches can be modelled as follows:

c1=s1, s2, ...sn
c2=s1, s2.....sn
cn=sn1, cn2, ...sn

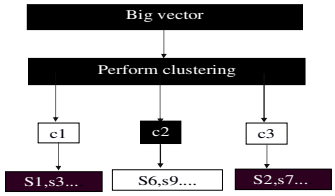


Figure 3 Representation of clusters

D. Generating Ranked Sentences

The best term from each county is obtained by applying statistical characteristics to each one once clustering has been completed. Since the statements in the clusters are initially expressed as numbers, we first reversal map the sentences from numbers to strings. This is accomplished by applying multiple statistical characteristics to the cluster, determining the rank of the utterances, and choosing the cluster's highest-ranking sentence. Pre - processing the language first, then employing statistical characteristics, is how the statistical features are used.

The pre-processing task is carried in 4 stages:

1) *Segmentation:* We may divide the document into paragraphs, paragraphs into phrases, and syllables into words by segmenting it..
2) *Synonym removal:* At this point, identical words are swapped out for a single syllable...
3) *Removing Stop Words:* Stop words such articles (a, an, the), prepositions (under, behind, etc.), and other common words that do not significantly contribute to the definition of the text's relevance are eliminated at this step and are not deemed significant enough to be included in the summaries. TF-IDF can help with this.
4) Word Stemming: The word's affixes, such as "s," "ing," and "ed," are ignored at this step, and only the root word is retained. This is conducted so that all of the terms that share the root word have the same term frequency.

After pre-processing, each sentence has the quantitative attributes performed, and the phrases are ranked.

a) Statistical Features

- *Sentence Length:* It is regarded as a crucial component that determines whether a sentence will be included in the description. The sentences that are longer are given more importance because they are seen to be more significant and relevant, and as a result, they are listed in the summary, whereas the statements that are shorter are ranked lower because they are not deemed to be as significant.
- Sentence Position: It is a crucial factor in determining whether or not to include the word in the analysis. The paragraphs that are written first are given more weight since they are seen as significant. The equation is used to determine the sentence's place

(1) Sentence Position= $1-\frac{S_i-1}{N}(1 < S_i < N)$ (1) where Si= the sentence number and,and, N= total number of sentences

- *Frequency (TF-IDF):*. We may use it to determine the frequency of terms used in the text and to compute their regularity in other papers. Based on that, we may determine if the word is significant or simply a common word that doesn't require much emphasis. In TF-IDF, which stands for "term frequency" and "inverse document frequency," each phrase's frequency is first determined, and then it is contrasted with the frequency of that word in unrelated documents. If a phrase appears often in other papers, it is deemed to be a common word and is excluded from the analysis. For example, words like 'from', 'a', 'an', 'the' etc are quite often present in the text document and their frequency would generally be high in a document.
- *Noun phrase and Verb phrase*: The strategy describes the significant statements that contain a noun or a group of words.
- *Proper Noun*: Emphasis is placed on the sentences that contain proper nouns. These are scored highly because they are thought to be significant. Using the POS tagger, the prepositional phrases are identified.
- *Aggregate Cosine Similarity*: It aids in determining whether or not the two sentences are comparable. Sentences are broken apart and represented as vectors in this way.

- *Cue Phrases*: In the paper, there are few sentences that are highlighted. For instance, the words "most notably," "although," "significantly," etc. are considered to be significant and include them in the report.

b) Semantic or Emotion features

The instinctual or intuitive sensation, as opposed to thinking or information, is what is referred to as sentiment. Emotions are a crucial component of human intellect, logical decision-making, social interaction, perception, memory, learning, and creativity since they are what define a human being and without them, there appears to be no difference between a man and an animal.

The eight sub-classes of emotions are: trust, anticipation, sadness, anger, joy, surprise, hatred, and disgust. These are then categorized into the two primary categories of positive and negative emotions. While the negative class comprises the emotions of sadness, hatred, rage, and disgust, the positive class contains the emotions of pleasure, trust, eagerness, and excitement.

c) Normalizing Values And Finding Total Score

In this step, the values are normalized or scaled to fall between 0 and 1 or -1 and 0. Additionally, normalization is done to convert standardized values from many scales to a single, universal scale. These characteristics are normalized as follows: -:

- *Normalizing Sentence Length Values:* The normalized sentence length can be computed as:

$$sLen_i' = \frac{sLen_i}{sLen_{max}} \qquad (2)$$

Where *s Leni* = sentence length of the *ith* sentence.

$sLen_{max}$ =sentence length value of the sentence having maximum sentence length value.

And $sLen_i'$ = is the normalized sentence length value of the *ith* sentence.

- *Normalizing Frequency (TF-IDF):* The TF-IDF normalized value is calculated by the following equation:

$$(tf * idf)_i' = \frac{(tf * idf)_i}{(tf * idf)_{max}} \qquad (3)$$

Where $(tf * idf)_i$ = term frequency-inverse document frequency value of the ith sentence.

$(tf * idf)_{max}$ = term frequency-inverse document frequency value of the sentence having maximum term frequency-inverse document frequency value.

And $(tf * idf)_i$' = normalized term frequency-inverse document frequency value of the ith sentence.

After normalization, we sum the values we acquired for each characteristic to determine the overall score for each phrase. The ranking of a sentence is determined by its overall score. The extract summary is chosen from these sentences based on rank, with higher ranks having a larger probability of being chosen. When creating a summary of n sentences, we pick the first n sentences (i.e the first n-sentences which are ranked the highest).

Calculating the overall score for the stated sentence results in:

$$TotalSore(s_i) = position_i + sLen_i' + (tf * idf)_i' + nvp_i' + PN_i' + ASC(S_i)' + CP_i' + emo_i' \qquad (4)$$

d) Redundancy Removing

There are frequently phrases with the same meaning but various wordings. To eliminate repetition, only one statement is picked from a group of lines with comparable meanings. Making use of the clustering algorithm on the other hand, items would be deleted and excluded from the breakdown if the cosine value is lower than the set given threshold.

e) Dealing With Connecting Words

There are some words in natural languages known as linking words, such as although, but, nonetheless, that if any statement began with them, their meaning is unclear alone without preceding sentence. As a result, our technology has been programmed to include the preceding phrase regardless of its rank if any sentence chosen for the final summary starts with any of these terms.

f) Making Abstract Summary

Our team has created a brand-new linguistic converter that combines WordNet, the Lesk automated system, and Parts-of-Speech tagger. We are attempting to create abstract summaries from the retrieved report using this language generators. Words are chosen when the extract summary is provided to the language generator so that they may be replaced with suitable synonyms to make it abstract. We obtain the Sunsets for a given word using WordNet. Then, a synonym for the term to be substituted is obtained using the Lesk method for word-sense disambiguation.

The part of speech of the term that Lesk produced is verified. The originating word is swapped by the recovered word if its Pos tag satisfies the POS of the item to be modified; otherwise, the method is restarted until the right substitution is made. The topology for getting an aggregate summary is shown in Fig. 4..

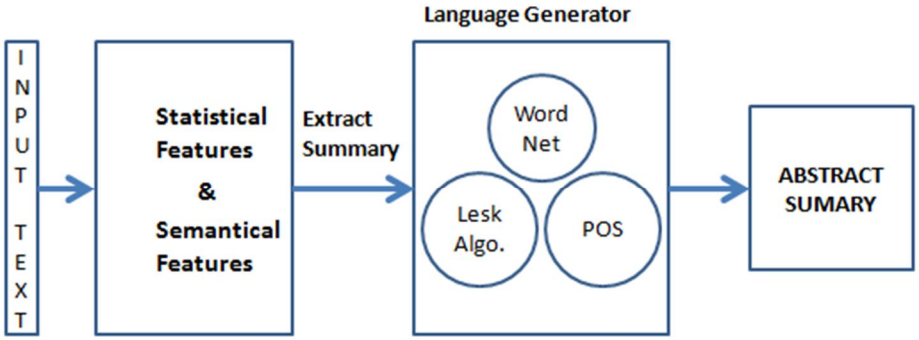


Figure 4 Making abstract summary

V. SYSTEM ARCHITECTURE

A. *Analysis And Algorithm Used*

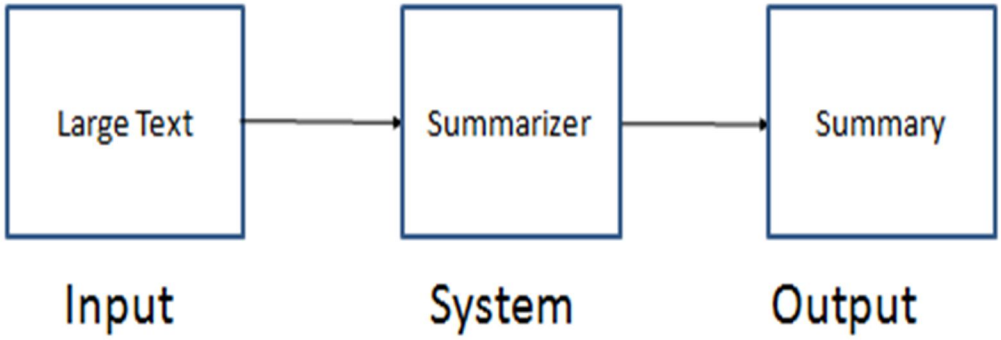


Figure 5 Block diagram of the summarizer

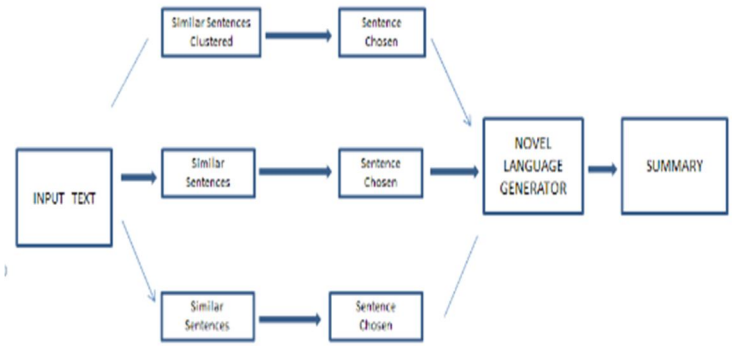


Figure 6 Block diagram of our summarizer

The algorithm used our summarizer is:
PASS 1: Pre processing
Input: Text Documents
Output: Pre-processed Text
PASS 2: Get Semantics Relationship Between the Sentences
Input: Pre-processed Text
Output: Semantically related sentences.
Step 1: use Word2vec to compute nearest words to each tokens in the sentences
Step 2: create big vectors to each sentence
PASS 3: Perform Clustering
Input: Big vector of sentences
Output: clusters based on similarity
Step 1: use k-Map clustering to perform clustering
PASS 4: Generating Ranked Sentences.
Input: Clusters
Output: Ranked Sentences
Step 1: Score the sentence given with 7different measures.

- Sentence Length
- Sentence Position
- Term Frequency-Inverse Document Frequency $w\ i = Tf * Idf = C(w) * \log(D/d(w))$

Where,
W i = the importance or weight of ith word,
C(w) =frequency of the current word In given document,
D= number of documents in the background corpus,
d(w)=number of background documents containing current word.
Step 2: Add a boost factor to those terms which are appear in capital.
Step 3: Rank the individual sentences according to them Weight value, pos values, boost factor, length of sentence and position of sentences
Step 4: Extract the higher ranked sentences of the input text in order to find the required summary.
PASS 5: Check for connecting words
Input: Ranked sentences.
Output: sorted ranked sentences
Step 1: check if any sentences in clusters begin with connecting words include previous sentences in the ranked sentences.
PASS 6: Using Cosine Similarity to Remove Redundancy.
Input: Sorted sentences.
Output: Salient sentences.
Step 1: Sorted sentences
Step 2: Summary = sentence having highest rank
Step 3: For i=1 to (total sentences) if [Similarity (Summary, i^{th} sentence) $<\theta$]
Then Summary= Summary + i^{th} sentence
Step 4: In order to uphold the sequence, rearrange the sentences according to their initial index.
PASS 7: Making Abstract Summary.
Input: Extracted Salient sentences.
Output: Abstract Summary.
Step 1: extracted sentences are fed to the novel language generator to transform them into Abstract summary.
PASS 8: Evaluation of summary
Step 1: Generate different summaries using Micro, Oponisis

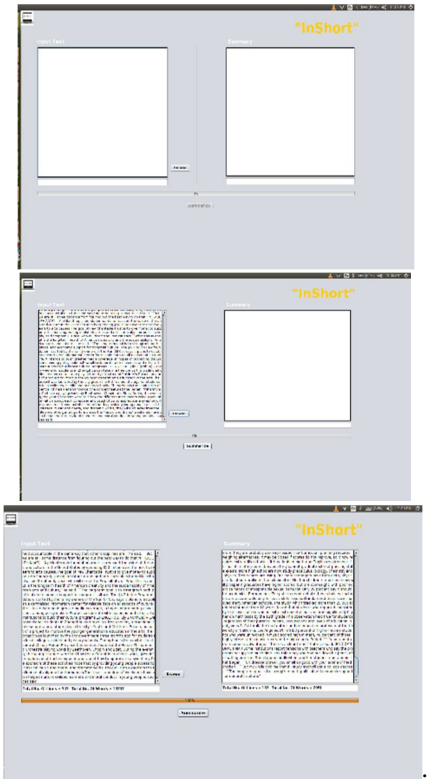
Auto Summarizer and our proposed Algorithms
Step 2: Use ROUGE to find Precious, Recall, and F-Score.

B. Corpus Description

The National Institute of Standards and Technology (NIST) has performed a series of summary evaluations known as the Document Understanding Conference (DUC) (NIST). DUC 2007 is the dataset utilized in this study. 43 papers will be included in DUC 2007. Both system and reference summaries are included in these papers.
Four reference reports and one system summary for evaluation are included in each document.
We are utilizing the ROUGE assessment software to assess the summary. DUC has selected ROUGE as its official assessment metric for both the summarizing of single-text documents and the summarization of multiple documents.

VI. EXPERIMENTAL RESULTS AND COMPARISON

In this project, several platform summaries are examined in relation to. other summaries are system produced summaries (i.e., Microsoft, Opinosis, Our- Algo, and Auto Summarizer are examples of system generated abstracts), whereas the Model (Gold/ Reference) summary is human developed.
In the first research, the algorithm that is used in this project, just generates a system overview using extractive characteristics. Tables 1 provide the results for a summary of 25% of the length reflecting various ROUGE scores.
The snapshots of our summarizer are as under:



Experiment 1
A hybrid summarization method was tested utilizing data from (DUC, 2007). Using the (DUC, 2007) datasets, certain articles were addressed. This algorithm creates a description for each internal representation around 55% of the original. The Microsoft Opinosis, Auto-Summarizer and the summaries produced by our method were compared
Three items have been measured in rough Recall, precision, and F-Score for every model summary (or reference summary) created by the method. ROUGE scores are determined using an algorithm.:
Precision = Count match (Sentence)/Count candidate (Sentence)
Re call = Count match (Sentence) / Count best sentence (Sentence)

Table 1 Summary generated by different systems and its comparison

Training		Metrices	Rouge-1	Rouge-1	Rouge-1	Rouge-1	Rouge-1	Rouge-1	Rouge-1
MICROSOFT	25%	Recall	0.81743	0.84902	0.79419	0.76351	0.83991	0.67593	0.38794
		Precious	0.06387	0.04867	0.04762	0.10588	0.05278	0.11017	0.46135
		F-Score	0.11487	0.08985	0.1897	0.09932	0.18942	0.42147	0.45678
OUR-METHOD	25%	Recall	0.50893	0.51058	0.340745	0.56034	0.4387	0.5516	0.57902
		Precious	0.33827	0.27216	0.29027	0.36791	0.27219	0.20058	0.34337
		F-Score	0.40641	0.35506	0.31349	0.44418	0.33595	0.29419	0.4311
OPINOSIS	25%	Recall	0.16999	0.0179	0	0.02355	0	0.04798	0.06498
		Precious	0.40385	0.5	0	0.54545	0	0.54545	0.34043
		F-Score	0.23927	0.03457	0	0.04514	0	0.0882	0.10912
AUTO_SUMMARIZER	25%	Recall	0.78851	0.69421	0.78353	0.73104	0.82089	0.28085	0.84438
		Precious	0.07679	0.0996	0.05795	0.12442	0.06379	0.33937	0.0768
		F-Score	0.13994	0.17421	0.10792	0.21264	0.11838	0.30735	0.1408

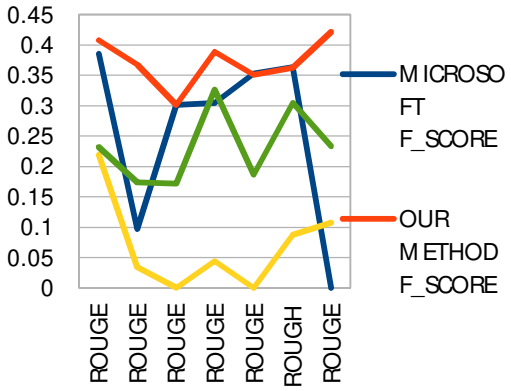


Figure 7 Showing different F-score curve (25%)

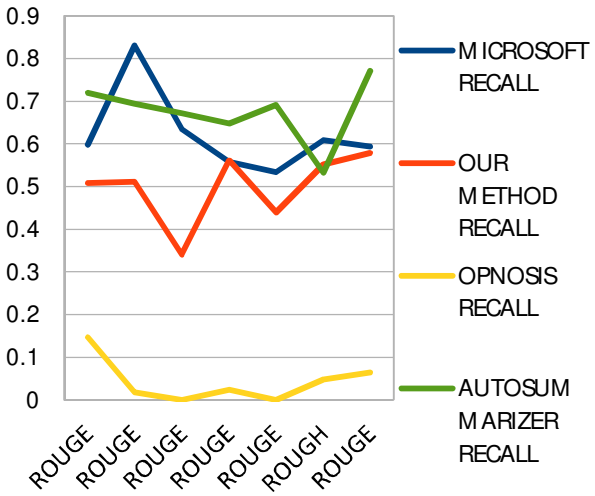


Figure 8 Showing different recall score curve (25%)

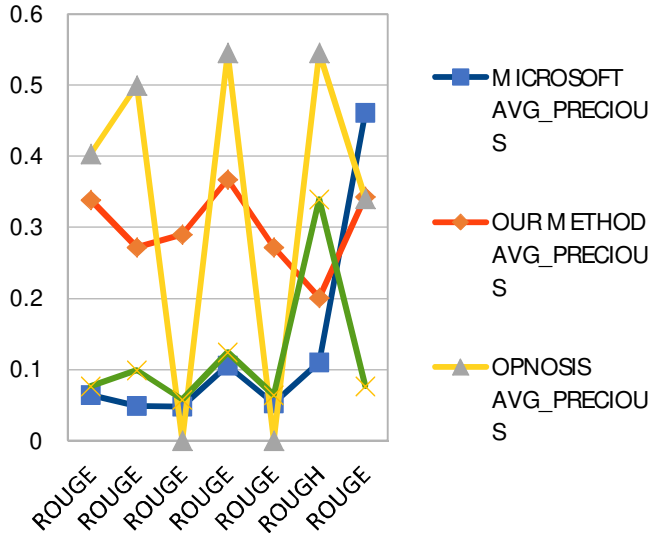


Figure 9 Showing different precision score curve (25%)

The cutting-edge speech synthesizer gets the constructed extracted analysis and transforms it into an abstractive analysis to make a hybrid display successful. It integrates WordNet, Lesk engine, and POS.

Comparative analysis of various semantic parameters shows that

1) In comparison to the precision score our method was in between opnosis and Microsoft

2) In terms of the recall, our method was way better than the opnosis .

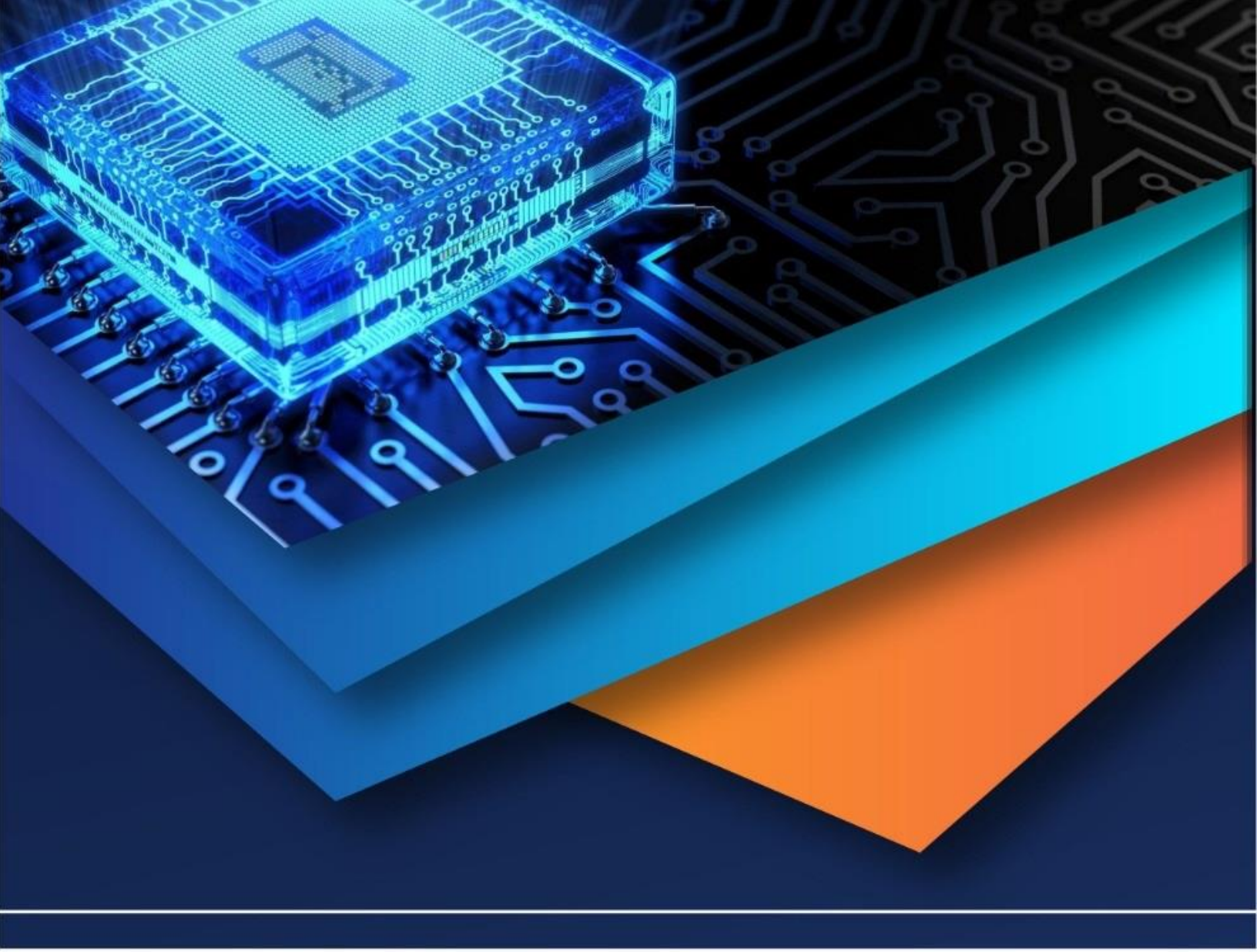
3) And this model is at par when it comes to F scores

VII. CONCLUSION, SUMMARY AND FUTURE SCOPE

In this study, we described a hybrid method for summarizing a single document. Our strategy is a fusion of abstraction and deduction. Utilizing Word2Vec deep learning, we first produced an extracted summarization using empirical and original semantic data. Matching and clustering are made possible by the semantic characteristic, which causes comparable phrases to be gathered. Once the integrative summary has been created, it is given to the innovative language compiler, where it is converted into the input text summary. Our system may be expanded to provide non - linear and non-summarizing.

REFERENCES

[1] R. C. Balabantaray, D. K. Sahoo, B. Sahoo, and M. Swain, “Text Summarization using Term Weights,” Int. J. Comput. Appl., vol. 38, no. 1, pp. 10–14, 2012.
[2] Nenkova and K. McKeown, “Automatic Summarization,” Found. Trends® Inf. Retr., vol. 5, no. 3, pp. 235–422, 2011.
[3] J. Kupiec, et al., "A trainable document summarizer," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995, pp. 68-73.
[4] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of Research and Development, vol. 2, pp. 159-165, 1958.
[5] B. Larsen, "A trainable summarizer with knowledge acquired from robust NLP techniques," Advances in Automatic Text Summarization, p. 71, 1999.
[6] D. Das and A. F. Martins, "A survey on automatic text summarization," Literature Survey for the Language and Statistics II course at CMU, vol. 4, pp. 192-195, 2007.
[7] H. Saggion and T. Poibeau, "Automatic text summarization: Past, present and future," in Multi-source, Multilingual Information Extraction and Summarization, ed: Springer, 2013, pp. 3- 21
[8] P.E. Genest and G. Lapalme, "Framework for abstractive summarization using text- to-text generation," in Proceedings of the Workshop on Monolingual Text-To-Text Generation, 2011, pp. 64-73.
[9] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries,” Proc. Work. Text Summ. branches out (WAS 2004), no. 1, pp. 25–26, 2004.
[10] Edmundson H, Wyllys R,: Automatic Abstracting and Indexing—Survey and Recommendations., Communications of the ACM., 4(5) (1961) 226-234
[11] Baxendale, P. B. (1958). Machine-made index for technical literature: An experiment. IBM Journal of Research and Development, 2(4), 354–361. doi:10.1147/rd.24.0354
[12] Kulkarni A R,: An automatic Text Summarization using feature terms for relevance measure. December 2002.
[13] R. Ferreira, L. De Souza Cabral, R. D. Lins, G. Pereira E Silva, F. Freitas, G. D. C. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro, “Assessing sentence scoring techniques for extractive text summarization,” Expert Syst. Appl., vol. 40, no. 14, pp. 5755–5764, 2013



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY