International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

AptShorts – A TEXT SUMMARIZER

Maleeha Arif Yasvi

Department of Computer Science and Engineering
Islamic University of Science and Technology
Awantipora, Pulwama, Jammu&Kashmir

ABSTRACT

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization approach can be of two types: Extractive and Abstractive. Our system is concerned with the hybrid of both the approaches. In our approach, we have used various statistical and semantic features to obtain the extractive summary. Emotions play an important role in one's life as emotions describe the state of our mind. So, we have used emotions as our semantic feature. On the basis of the emotions described by the writer in the document, we have classified the summaries into two- summary with only positive content and the summary with only negative content. The generated extractive summary is then passed to our novel language generator which constitutes WordNet, Lesk algorithm and part-of-speech tagger that transforms the extractive summary into abstractive one, resulting in the hybrid summarizer. We have evaluated our summary using DUC-2007 data set and achieved significant results as compared to the MS-Word summarizer.

Keywords: Abstract Summary, Extract Summary, Hybrid Summarization, Semantic Features, Statistical Features, WordNet

I INTRODUCTION

The explosion of electronic documents has made it difficult for users to extract useful information from them. The user due to large amount of information does not read many relevant and interesting documents. As such, automatic document summarization is an essential technology to overcome the obstacle. Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in large number of sectors in industry today. An example includes document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary or abstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important(or salient) images. Similarly, in consumer videos, one would want to remove the

boring or repetitive scenes, and extract out a much shorter and concise version of the video. Generally, there are two approaches to automatic summarization: Extraction and Abstraction.

1. Extraction-based summarization:

   Extractive methods work by selecting a subset of existing words, phrases or sentences in the original text to form the summary. In this summarization task, the automatic system extracts objects from the entire collection, without modifying the objects themselves.

2. Abstraction-based summarization:

   Abstractive method builds an internal semantic representation and then, uses natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Abstractive methods condense a text more strongly than extractive.

The work that has been carried out until now has mainly focused upon the extractive techniques for summarization. Very less work has been done on summarizing the text document using abstractive techniques.

The extractive technique generally includes mechanisms in which the salient sentences and paragraphs are chosen from original document and those sentences are concatenated to form the summary. Such features are referred as the statistical features.

Many statistical features are also used to derive extractive summary. These include:

Sentence Length, TF-IDF factor, Cosine Similarity, Noun Phrase and Verb Phrase, Sentence position etc.

The work that has been carried out in abstractive summarization involves the use of natural language processing (NLP) and machine learning. In this approach, the document selected for summarization is passed through various NLP techniques and machine learning techniques to generate a summary that is paraphrased and closer to what a human might generate.

Abstractive method builds an internal semantic representation and then uses NLP to create a paraphrased summary.

Our approach is concerned with the hybrid of both the techniques (extractive and abstractive).We have used both the abstractive and extractive approaches in the document to generate the hybrid summary. So far, summarization task has focused on only statistically significant features, but in our approach we have focused on semantically enriched learning and build a hybrid summarizer exploiting semantic learning and features. Emotions play an important role in one's life as emotions describe the state of our mind. So, we have used emotions as our semantic feature. On the basis of the emotion described by the writer in the document, we have classified the summaries into two- summary with only positive content and the summary with only negative content. By using these statistical and semantic features, we obtained extractive summary. The extractive summary generated is then passed to our novel language generator comprising of WordNet, Lesk algorithm and POS tagger. Hence, we obtain a hybrid summary.

II RELATED WORK

Sentence extraction came into existence in 1950s, when the automatic text summarization was started by Luhn. For identification of salient sentences, he used the frequency of occurrence of words; words that occur frequently in a document are important and should therefore be considered in the summary [2].

Edumpson modified Luhn's work by focusing on more features that could be considered in the summary. These include- word frequency, count of title words, sentence position and the number of cue-phrases occurring in the document [2]. Edumpson laid much stress on the sentence length (the total number of words constituting in a sentence) and sentence position (the position is also considered as an important factor,for example, the sentences written in the start are considered).

The work carried out focused on only the extractive techniques. Major breakthrough in abstractive techniques came when Paice submitted his work in language generation techniques in which he emphasized upon the natural language techniques and machine learning to obtain the abstractive summary [2].

Much amount of work is done in extractive techniques. The high priority sentences are selected from the document and are written as a summary of that document [3]. The sentences are selected by using various statistical features [4] and then, ranking of the sentences is done so that we obtain a précised summary of few lines based on the sentences chosen from the document [5].

In abstractive summarization, the sentences are not selected from the document but instead paraphrasing of sentences is done [6]. The summary formed by abstractive technique is quite closer to the one generated by humans. The abstractive summary is formed by generally using the language generation techniques [6].

On the basis of content, the summaries are of two types- indicative and informative [7]. The summaries that give a general idea about the input text are referred as the indicative summaries while as the summaries that give the whole information about the input text are referred as the informative summaries.

Initially, the summarization task was restricted to only single document i.e summary could be obtained of a single document only [3] but now, with the advancement of technology and research, multi-document summarization is possible. In multi-document summarization, single summary can be obtained for multiple documents.

Natural language processing (NLP) is used for obtaining a paraphrased summary in abstractive techniques of summarization [8].

For the evaluation of summary and to know the quality of summary, various metrics are implemented like- coherence, conciseness, grammar, readability and content [9]. For evaluation of the summaries, various evaluators are used such as ROGUE, MEAD, OPINOSIS, MS-WORD [9]. These help in evaluating the summary and checking the quality and precision of the summarizer.

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

III METHODOLOGY

For the summarization process, we use both the abstractive and extractive technique in order to obtain the summarizer that is efficient and the one that generates a summary closer to what a human might generate. Our summarization procedure is divided into 5 stages:

- Preprocessing
- Generating Ranked Sentences
- Normalizing values and finding total score
- Using cosine similarity to remove redundancy
- Making abstract summary

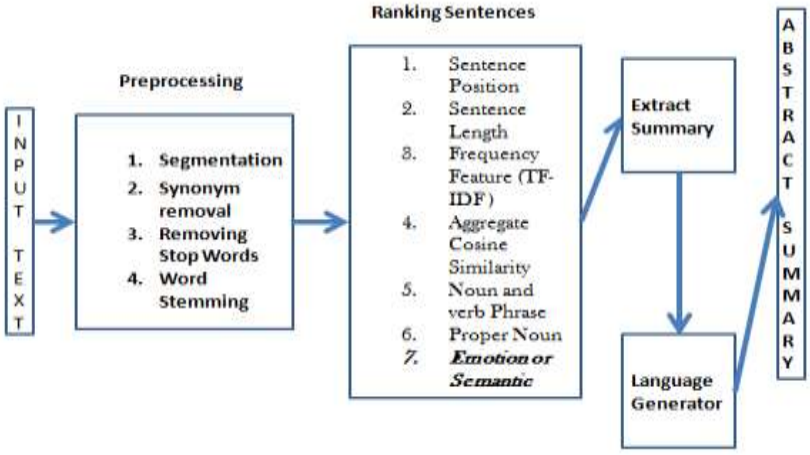


Fig 1. Diagrammatic representation of our system

3.1 Preprocessing:

Before performing the sentence scoring, we first preprocess the data. Preprocessing of the data improves the quality of data and hence improves the quality of summary too. The preprocessing task is divided into 4 stages:

1. *Segmentation*: The first stage of preprocessing is the segmentation. Segmentation allows us to fragment the document into paragraphs, paragraphs into sentences, sentences into words. This is done so as to obtain the tokens. Tokens are obtained so that they can be ranked.
2. *Synonym removal*: In this stage, the words with same meaning are replaced by a single word. This is done so that we obtain the correct term frequency of each word and do not take separate frequency for the words with similar meaning.
3. *Removing Stop Words*: In this stage, the stop words like articles(a, an , the), prepositions(under, behind, etc) and other common words that do not play any significant role in defining the importance of the text are removed and are not considered important to be the part of summary. This can be achieved by TF-IDF.

4. *Word Stemming*: In this stage, the affixes from the word like 's', 'ing', 'ed' are neglected and only the root word is kept. This is done so that we do not obtain different term frequencies for all the words with the same root word. So, stemming ensures single term frequency by considering only the root word. For example, the words like 'playing', 'plays', 'played' are replaced by a single root word 'play'. For word stemming, we use Porter Stemmer Algorithm.

3.2 Generating Ranked Sentences:

After preprocessing the sentences, we rank them based on the following statistical and semantic features:

-Statistical Features:

1. *Sentence Length*: It is considered as an important feature that decides whether to include the sentence in the summary or not. The sentences which are longer are considered to be important and are ranked higher, so they form a part in summary while the sentences which are shorter are ranked low, so not considered important and thus not included in the summary.

2. *Sentence Position*: It is an important parameter that decides whether to include the sentence in summary or not. The sentences which are written first are considered important , so these are ranked higher. The sentence position is calculated by using the equation (1)

$$\text{Sentence Position} = 1 - \frac{S_i - 1}{N} \qquad (1 < S_i < N) \qquad \text{(Er}$$

where Si= the sentence number and,

and, N= total number of sentences

3. *Frequency(TF-IDF):* It helps us to find the frequency of words occurring in the document and their relative occurrence in other documents is also calculated. On that basis, we can conclude whether the word is important or it is a common word and should not be emphasized much. In TF-IDF(TF stands for term frequency and IDF stands for inverse document frequency), first the frequency of each term is calculated and then, it is compared with the frequency of that term in other, non-relating documents. If the frequency of that term in other documents is high, then it is considered to be a common word and not included in the summary. 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. But, by using IDF, we can calculate its occurrence in other non-relating documents and decide whether to consider it as an important word for the summary or not.

TF-IDF is calculated by using the equation (2)

$$TF_i * IDF_i = f(w) * \log\left(\frac{bg}{bg(w)}\right) \qquad \text{(Er}$$

Here,

TFi =the term frequency of the ith word in the document.

IDFi = the inverse document frequency of ith word in the document.

f (w) is the frequency count of the ith word in given text

bg = total number of background documents taken.

bg (w) = is number of background documents that contain the i th word.

4. *Noun phrase and Verb phrase*: The sentences containing noun or a verb phrase are considered important and are included in the summary. The noun and verb phrases are tagged in the input text by using the Standford POS tagger. Tagging process is used to assign various parts of speech such as nouns, verbs, adjectives, determiners, noun phrases, verb phrases etc.

5. *Proper Noun*: The phrases in which proper noun is written are emphasized upon. These are considered to be important and hence are ranked high. The proper nouns are tagged by using the POS tagger.

6. *Aggregate Cosine Similarity*: It helps us to determine whether the two sentences are similar or not. It is done by fragmenting sentences and representing them in a vector form. After forming vectors, the cosine angle is fund between the two sentences which determines if the two are similar or not. If the two sentences are similar, then only one is chosen among the two and is included in the summary. And, if the two are different, then both are included in the summary. The implementation of cosine similarity is done by the equation (3)

$$sim\ (S_i S_j) = \sum_{k=1}^{n} W_{ik} * W_{jk} \qquad \text{(Er}$$

$$Aggregate\ Cosine\ Similarity(s_i) = \sum_{j=1, j \neq i}^{n} sim\ (S_i S_j) \qquad \text{(Er}$$

Where W_{ik} = TF-IDF of the k^{th} word in the i^{th} sentence.

The cosine similarity between two sentences can be represented as $S_i=[W_{i1},W_{i2},..W_{im}]$ and $S_j=[W_{j1},W_{j2},....W_{jm}]$; according to this rule cosine similarity between W_{i1} (first word of sentence "i") and W_{j1} (first word of sentence "j") is calculated and similarly for the rest words of the sentence. The sum of cosine similarity measure for words present in the sentence represents the cosine similarity of that particular sentence. Our implementation uses the below mentioned rule which represents the Standard Cosine Similarity.

$$sim\ (S_i S_j) = \frac{\left(\sum_{k=1}^{n} W_{ik} * W_{jk}\right)}{\sum_{k=1}^{n} {W_{ik}}^2 \ \sum_{k=1}^{n} {W_{jk}}^2} \qquad i,j = 1 \text{ to } n \qquad \text{(Er}$$

7. *Cue Phrases*: There are certain phrases present in the document which are emphasized upon. For example, the phrases beginning with 'Most importantly', 'However', 'Significantly' etc are considered important and are included in the summary.

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

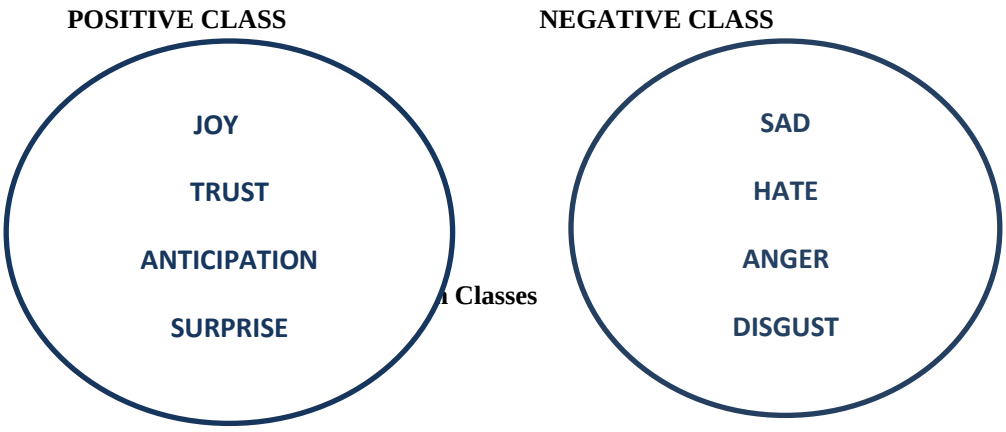
Semantic or Emotion features:

Emotion can be defined as the instinctive or intuitive feeling as distinguished from reasoning or knowledge. As emotions make a human being, without it, there seems no difference between a man and an animal, so emotions play an important role in human intelligence, rational decision making, social interaction, perception, memory, learning, creativity.

Emotions are also important in text summarization as these help in knowing the affinity of user and his perception over the writing. On the basis of emotion feature, we have classified the summaries into two types:

- The one in which only the positive part of the text will be included
- And the other, in which the negative part of the text will be included

We have divided the emotions into 8 sub-classes –: trust, anticipation, sad, anger, joy, surprise, hate and disgust and then we have grouped these into 2 main classes i.e positive and negative. The positive class contains the joy, trust, anticipation and surprise emotion while as the negative class contains the sad, hate, anger and disgust emotion.



Classes

If we want to include only positive words in the summary, we assign positive value for the words present in the positive class and negative value for the words present in the negative class. Similarly, if we want that the summary should include only negative part of text then we assign a positive value to the words present in negative class and negative value to the words present in positive class. So, in this manner we can obtain a much refined summary which will also focus on the emotions expressed by the writer.

The implementation is done using the WordNet Affector that is a subset of WordNet in which all the synonyms of emotion words are present in the respective class.

3.3 Normalizing values and finding total score:

In this stage, we normalize or scale the values so that they range between 0 to 1 or -1 to 0. Normalization is also performed so that standardized value is obtained from different scales and is thus normalized to one common scale. The normalization of the following features is done-:

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

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

Where *sLeni* = 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:

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

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.

-*Normalizing Noun phrase and Verb phrase:*The normalized value of noun and verb phrase can be calculated as:

$$nvp_i' = \frac{nvp_i}{nvp_{max}} \qquad (Er$$

Where nvp_i = noun-verb phrase value of the i^{th} sentence.

nvp_{max} = noun-verb phrase value of the sentence having maximum noun and verb phrase value.

And nvp_i' = normalized noun-verb phrase value of the i^{th} sentence.

Normalizing Proper noun value: The normalized value for proper noun calculation is as:

$$PN_i' = \frac{PN_i}{PN_{max}} \qquad (Er$$

Where PN_i = proper noun value of the i^{th} sentence.

PN_{max} =proper noun value of the sentence having maximum proper noun value.

And PN_i' = normalized proper noun value of the i^{th} sentence.

– *Normalizing Aggregate Cosine Similarity Values:* Value for the aggregate cosine similarity feature of each sentence is normalized by computing

$$ACS(S_i)' = \frac{ACS(S_i)}{ACS(S_{max})} \qquad (Err$$

Where $ACS(S_i)$ = aggregate cosine similarity of the i^{th} sentence.

ACS(S_{max}) = aggregate cosine similarity value of the sentence having maximum aggregate cosine similarity value. And *ACS(S_i)'* is the normalized aggregate cosine similarity value of the i^{th} sentence.

– *Normalizing Cue-phrases Values:* Normalized values for the cue-phrase feature of each sentence is obtained by using the below mentioned formula,

$$CP_i' = \frac{CP_i}{tCP} \qquad \text{(Err}$$

Where, CP_i = Count of the cue-phrases in the i^{th} sentence.

tCP = Total number of cue-phrases in the text document.

And CP_i' = Normalized cue-phrase score of the i^{th} sentence.

– *Normalizing Emotion Values:* Value for the emotion feature of each sentence is normalized by computing

$$emo_i' = \frac{emo_i}{emoT} \qquad \text{(Err}$$

Where, emo_i = Number of emotion words in the i^{th} sentence.

emoT = Total number of emotion words in the text document.

And emo_i' = Normalized emotion score of the i^{th} sentence.

After normalization, we calculate the **total score** for each sentence by adding the values obtained for every feature. This total score for a given sentence represents its **rank.** On the basis of this rank sentences are opted for the extract summary, higher the rank greater are the chances for selection. If we have to form a summary of n-sentences, then we choose the first n-sentences (i.e the first n-sentences which are ranked the highest).

The total score for the given sentence is obtained by computing:

$$TotalSore(s_i) = position_i + sLen_i' + (tf * idf)_i' + nvp_i' + PN_i' + ASC(S_i)' + CP_i' + emo_i' \qquad \text{(Error! Bookmark not defined.)}$$

3.4 Redundancy removing:

Many a times, there are similar meaning sentences with different wordings. So, among all the similar meaning sentences, only one is chosen out of the lot in order to avoid redundancy. This is done by using the cosine similarity. A particular threshold value is set. If the cosine similarity of the sentences is above that threshold then, these will be included in the summary. On the other hand, if the cosine value is smaller than the predefined threshold value, then these would be discarded and not included in the summary.

3.5 Making Abstract Summary:

After obtaining the extractive summary (by analyzing the statistical features and then , ranking them) , we then obtain the abstractive summary. The data obtained from extractive technique is passed to our uniquely designed Novel Language Generator. This Novel language generator consists of WordNet, Lesk algorithm and POS tagger. WordNet is used so that we can obtain the similar / related words of a particular word. Lesk algorithm is used so that we obtain only the similar meaning words of a particular word . POS tagger is used so that the word if replaced by any other synonym word belongs to the same part of speech as that particular word.

Fig 3 represents the block diagram of our Novel Language Generator

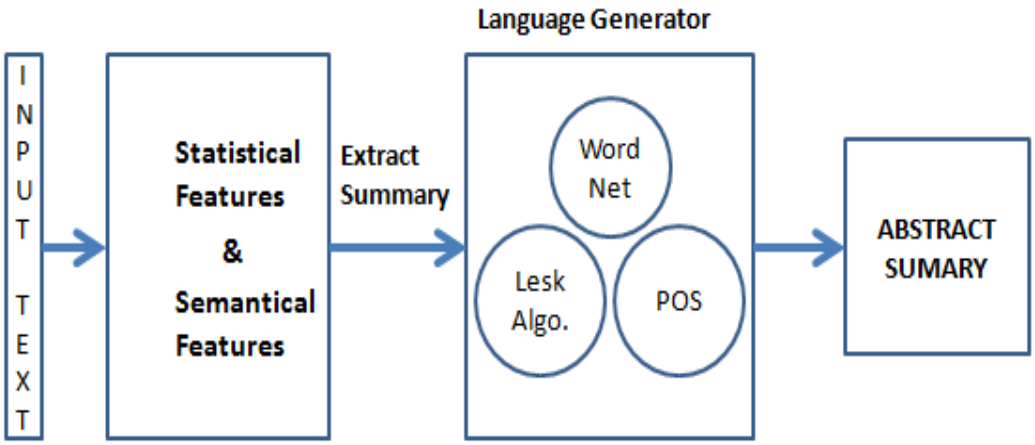


Fig 3. Block diagram of Novel Language Generator

Algorithm:

Stage 1: Preprocessing

Input: Text Documents

Output: Preprocessed Text

Segmentation, Synonym removal, Removing Stop Words

Stage 2: Generating Ranked Sentences.

Input: Preprocessed Text

Output: Ranked Sentences

Step 1: Score the sentence given with 8 different measures.

Sentence Length, position, TF-IDF, noun-verb phrase ,proper noun ,cosine similarity ,cue-phrases ,emotion score

Stage 3: Normalizing Values And Finding Total Score.

Input: Values That are Not Normalized.

Output: Normalized Values And Their Sum In Sorted Order.

Step 1: Apply normalization rules.

Normalized Sentence Length, sentence position ,TF-IDF ,noun phrase, proper noun ,cosine similarity ,cue-phrases ,emotion score

On the basis of emotion score and the type of text we want to include in our summary, we can assign positive value to either the positive or negative class(depending upon the emotion words we want to include in our summary) and thus obtain a more refined summary.

Step 2: Add all the features for every sentence. This sum represents the rank of the given sentence.

Step 3: Sorting on the basis of *Total Score*, sentence having the highest total score is the most important sentence.

Stage 4: Using Cosine Similarity To Remove Redundancy.

Input: Sorted sentences.

Output: Salient sentences.

Step 1: Summary = NULL and Threshold value is 0.15.

Step 2: Summary = sentence having highest rank

Step 3: For i=1 to (total sentences) if [Similarity (Summary, i^{th} sentence) <θ]

Then Summary= Summary + i^{th} sentence

Step 4: In order to uphold the sequence, rearrange the sentences according to their initial index.

Stage 5: Making Abstract Summary.

Input: Extracted Salient sentences.

Output: Abstract Summary.

Extracted sentences are fed to the novel language generator to transform them into Abstract summary.

IV ANALYSIS AND RESULTS

We evaluated our document hybrid summarization system using (DUC, 2007) dataset. We took 15 news articles as input. For each input document, our system generated summary which was about 35% of the original document. We compared our system generated summaries with the MS-Word generated summaries. The evaluation of our summarizer was done using ROGUE.

Table1. Comparison of the summaries generated by our system and MS-Word

Input Document	Sentences Selected by our system	Sentences Selected by MS Word	Sentences Selected by Human Judgement	Precision of MSWord	Precision of our system
InputDoc1	1,3,4,9,12	1,3,5,10,12	1,2,3,9,14	0.4	0.6
InputDoc2	2,4,5,6,10	1,4,5,8,12	1,2,4,5,15	0.6	0.6
InputDoc3	1,6,9,12,19	2,5,9,13,19	1,6,9,13,20	0.4	0.4
InputDoc4	1,8,9,13,19	1,9,13,14,16	1,8,11,13,19	0.6	0.4
InputDoc5	1,6,8,10,14	3,6,8,14,20	2,6,10,14,20	0.6	0.6
InputDoc6	1,2,7,10,13	2,5,8,12,14	1,2,7,13,18	0.2	0.8
InputDoc7	1,2,4,11,16	4,10,11,15,19	1,2,4,11,19	0.6	0.8
InputDoc8	1,4,11,12,20	1,3,8,12,20	1,2,3,4,20	0.6	0.6
InputDoc9	1,2,16,18,19,20	1,8,13,14,17	1,4,8,14,19	0.4	0.6
InputDoc10	1,3,5,8,11	5,8,9,10,12	1,2,3,8,14	0.2	0.6
InputDoc11	3,4,10,12,19	3,7,10,12,18	3,4,7,15,19	0.6	0.4
InputDoc12	1,3,6,8,13	4,10,12,14,16	1,3,6,16,19	0.2	0.6
InputDoc13	1,2,6,10,18	1,5,9,12,19	2,6,9,12,18	0.4	0.6
InputDoc14	1,2,3,7,17	1,2,6,15,18	1,2,3,14,18	0.4	0.4
InputDoc15	1,3,5,13,15	3,5,11,16,18	1,3,5,12,15	0.4	0.8

ROUGE-Ty	Task Name	System Na	Avg_Recal	Avg_Precis	Avg_F-Sco	Num Refer
ROUGE-1	D0719	SYSTEM19	0.60699	0.04776	0.08854	4
ROUGE-1	D0718	SYSTEM18	0.67735	0.01578	0.03084	4
ROUGE-1	D0717	SYSTEM17	0.75547	0.04869	0.09149	4
ROUGE-1	D0739	SYSTEM39	0.5087	0.03026	0.05712	4
ROUGE-1	D0716	SYSTEM16	0.5112	0.0603	0.10788	4
ROUGE-1	D0738	SYSTEM38	0.59975	0.22972	0.3322	4
ROUGE-1	D0715	SYSTEM15	0.58246	0.01286	0.02516	4
ROUGE-1	D0737	SYSTEM37	0.52723	0.01229	0.02402	4
ROUGE-1	D0714	SYSTEM14	0.62987	0.01489	0.0291	4
ROUGE-1	D0736	SYSTEM36	0.55	0.02912	0.05532	4
ROUGE-1	D0713	SYSTEM13	0.6064	0.01407	0.02751	4
ROUGE-1	D0735	SYSTEM35	0.64849	0.01506	0.02943	4
ROUGE-1	D0734	SYSTEM34	0.7562	0.04898	0.09199	4
ROUGE-1	D0712	SYSTEM12	0.63356	0.01563	0.0305	4
ROUGE-1	D0733	SYSTEM33	0.67975	0.01594	0.03114	4
ROUGE-1	D0711	SYSTEM11	0.76201	0.01839	0.03592	4
ROUGE-1	D0710	SYSTEM10	0.62398	0.01583	0.03088	4
ROUGE-1	D0732	SYSTEM32	0.49659	0.16008	0.24211	4
ROUGE-1	D0731	SYSTEM31	0.69896	0.01645	0.03215	4
ROUGE-1	D0730	SYSTEM30	0.71014	0.09368	0.16553	4
ROUGE-1	D0709	SYSTEM9.1	0.72489	0.01944	0.03786	4
ROUGE-1	D0708	SYSTEM8.1	0.53069	0.01737	0.03364	4
ROUGE-1	D0729	SYSTEM29	0.65314	0.01476	0.02887	4
ROUGE-1	D0707	SYSTEM7.1	0.64533	0.05517	0.10165	4
ROUGE-1	D0706	SYSTEM6.1	0.55812	0.13784	0.22108	4
ROUGE-1	D0728	SYSTEM28	0.66896	0.01429	0.02799	4
ROUGE-1	D0705	SYSTEM5.1	0.49369	0.16576	0.24819	4
ROUGE-1	D0727	SYSTEM27	0.66484	0.01544	0.03018	4
ROUGE-1	D0740	SYSTEM40	0.62712	0.10396	0.17836	4
ROUGE-1	D0704	SYSTEM4.1	0.60662	0.0144	0.02813	4
ROUGE-1	D0726	SYSTEM26	0.70819	0.01734	0.03385	4
ROUGE-1	D0703	SYSTEM3.1	0.63322	0.01529	0.02986	4
ROUGE-1	D0725	SYSTEM25	0.62741	0.21174	0.31663	4
ROUGE-1	D0702	SYSTEM2.1	0.62639	0.01426	0.02789	4
ROUGE-1	D0724	SYSTEM24	0.66156	0.01655	0.03229	4
ROUGE-1	D0723	SYSTEM23	0.74385	0.0405	0.07682	4
ROUGE-1	D0701	SYSTEM1.1	0.71616	0.02695	0.05195	4
ROUGE-1	D0745	SYSTEM45	0.70228	0.06267	0.11507	4
ROUGE-1	D0744	SYSTEM44	0.69712	0.01659	0.03242	4
ROUGE-1	D0722	SYSTEM22	0.50347	0.23966	0.32474	4
ROUGE-1	D0743	SYSTEM43	0.66326	0.01555	0.03039	4
ROUGE-1	D0721	SYSTEM21	0.71536	0.02441	0.04722	4
ROUGE-1	D0720	SYSTEM20	0.66533	0.01527	0.02986	4
ROUGE-1	D0742	SYSTEM42	0.04531	0.41964	0.0818	4
ROUGE-1	D0741	SYSTEM41	0.81625	0.02192	0.0427	4

Table 2. Evaluation done by ROGUE

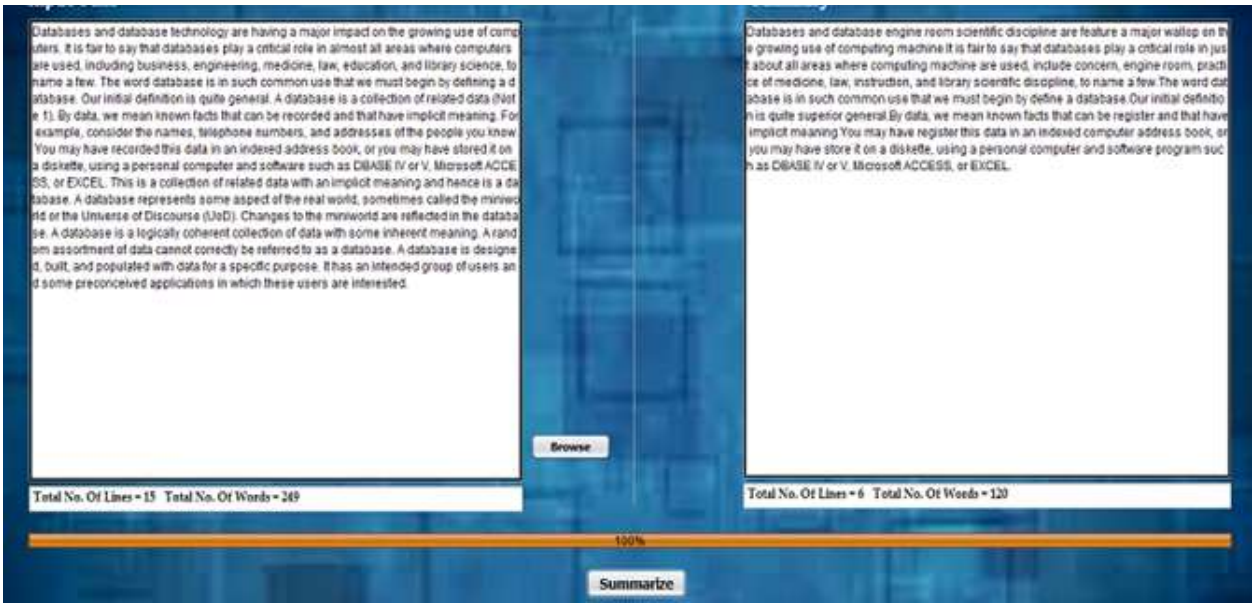


Fig 4. Our Hybrid Summarizer

V CONCLUSION AND FUTURE WORK

In this paper, we explained a hybrid approach to document summarization. Our approach was the hybrid of both the techniques- extraction and abstraction. We first generated extract summary using statistical and novel semantic (emotion) features. We used emotion feature as emotion plays a critical role in identifying the salient sentences to be included in the extract summary. Then using novel language generator extract summary was transformed into abstract summary. We evaluated our system by comparing its summaries and MS-Word generated summaries with the human generated summaries and achieved significant results. In most of the cases the relevance rate of our system was more than MS-Word. We also evaluated our system using ROGUE.

Further, we can enhance our system by using various machine learning algorithms and neural networks to obtain a more précised and refined summary.

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—Surveyand Recommenda- tions., 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
14. Gupta, P., Pendluri, V. S., & Vats. I. (2011). Summarizing text by ranking text units according to shallow linguistic features. In 13th International conference on advanced communication technology (pp. 1620–1625).
15. Prasad, Rajesh Shardanand, Uplavikar, Nitish Milind, Wakhare, Sanket Shantilalsa, Jain, Vishal, Yedke & Tejas Avinash (2012). Feature based text summarization. International Journal of Advances in Computing and Information Researches, 1.
16. Kulkarni, U. V., & Prasad, Rajesh S. (2010). Implementation and evaluation of evolutionary connectionist approaches to automated text summarization. In Journal of Computer Science (pp. 1366–1376). Science Publications.