Automatic Text Summarization Using Local Scoring and Ranking

[1]Diksha Kumar, [2]Sumeet Bhalekar, [3]Hari Disle, [4]Sameer Gorule, [5]Ketan Gotarane

[1]Faculty, [2]Student, [3]Student, [4]Student, [5]Student
Department of Information Technology,
Saraswati College of Engineering, Kharghar, India

***Abstract*: Existence of large amount of textual information available on the internet emerged serious research in the area of machine generated summarization. Manual summarization of these online text documents is a very difficult task for human beings. So we need an automatic text summarizer. Automatic Text Summarization (ATS) is "condensing the source text into a shorter version, while preserving its information content and overall meaning". Even though the work of automatic text summarization started in 1950's, still it is lacking to achieve more coherent and meaningful summaries. The proposed approach provides automatic feature based extractive text summarizer to improve the coherence thereby improving the understandability of the summary text. It summarizes the given input document using local scoring and local ranking that is it provides summary. The proposed approach applies the features to all document sentences. But it ranks the sentences and selects top n sentences from each paragraph where n depends upon compression ratio. The final summary produced by this approach is a collection of summary of individual paragraph/heading. Since the summary contains the equal proportion of sentences from each heading, it reduces the coherent gap of the summary text. Also it improves the overall meaning and understanding of the summary text.**

***Keywords*: Text summarization; manual summarization; compression ratio; main summary; coherent gap; precision**

I. INTRODUCTION

Nowadays automatic text summarizer is a dominant automated software tool for processing large amount of online information. Its goal is producing a shorter text for the given text without loss in overall information of the source text. With the help of this software tool, people understand more number of documents in short period and decide whether to read the entire document or not. ATS process should address the problems of selecting salient portions of text and creating coherent summaries. Generally automated summaries differ more from that of human generated summaries because of human reasoning.

The two kinds of text summarization methods are abstractive summarization and extractive summarization. An abstractive summarization process produces the summary text by rephrasing the original text after understanding it clearly. An extractive summarization process produces the summary text by selecting salient portions of the text. The salient portions may be word, or Sentence or phrase or paragraph. Sentence importance is decided by the features used in the summarization process. In this work, we introduce automatic summarizer to produce extractive summary[1].

Summaries make the task of understanding the meaning of text easier. Text summarization helps user to manage vast amount of information by condensing document and include more relevant facts into them. Text summarization process contains three steps: analysis, transformation and synthesis. The general steps of text summarization or ATS. The input of the system can be single or multiple documents. It depends on the user requirement. The next step is Pre-processing in this step stop words removed and tokenization performed. Sentence Analysis step includes sentence scoring and sentence ranking to rank the sentence. From this, the final summary is generated which is the final and the last step of the system[2].

II. SYSTEM DETAILS

Our proposed work is the generation of extractive single document summarizer to improve the coherence of the summary text. Extractive approach works by selecting important sentences. Numerous methods are used for sentence selection. The most widely used method is sentence scoring method. This method assigns some numerical value to a sentence by summing all the feature values of the sentence. Finally, it ranks the sentences and selects top n sentences, where n depends upon compression ratio. In this section, we produce two kinds of summaries: Heading wise summary and Main summary[1]

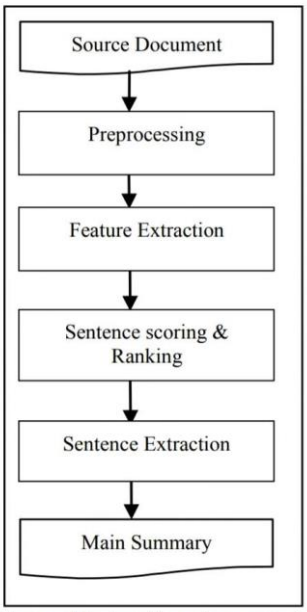
Main summary generation It produces summary text for the entire source document using the features listed below[1].

It has the following steps:

1. Give the document to be summarized as input.
2. Pre-process the input text. First sentence tokenize and word tokenize the input text. Then perform stopping and stemming operations.
3. Calculate feature score and sentence score. Each sentence has six features F1, F2, F3, F4, F5, and F6 and a score for each feature. Here, the score for sentence resemblance to the heading (F5) is zero for all sentences.
4. Calculate sentence score and rank the sentences. Sentence scores are obtained by adding the feature scores.
5. Finally, select the top n sentences as the summary sentences, where the value of n depends on the compression rate.

6. Order the summary sentences according to the order of the sentences in the original text and display as output summary text. In the generation of Main summary, the score for a sentence Si can be calculated as follows:
As we have six features of sentences,
Score(Si) = F1(Si)+F2(Si)+F3(Si)+F4(Si)+F5(Si)+F6(Si)/ 6

III. PROPOSED SYSTEM



System architecture for main summarizer.

SOURCE DOCUMENT
Source document is an input file on which we perform text summarization.

PREPROCESSING
Step 1: Tokenization
What is tokenization?
Tokenization is a way to split text into tokens. These tokens could be paragraphs, sentences, or individual words.
How it is performed?
The java.util.StringTokenizer class allows an application to break a string into tokens.
This class is a legacy class that is retained for compatibility reasons although its use is discouraged in new code.
Its methods do not distinguish among identifiers, numbers, and quoted strings.
This class methods do not even recognize and skip comments.
Why it is used?
A StringTokenizer object internally maintains a current position within the string to be tokenized. Some operations advance this current position past the characters processed. A token is returned by taking a substring of the string that was used to create the StringTokenizer object.
Step 2: Stemming
What is Stemming?
Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form.
How it is performed?
In stemming we convert the noun, verb or adjective form of a word into the normal form. We do this by using a predefined library known as snowball. By converting a word into its normal form, summarization becomes easier. Thus we chose stemming for pre-processing.
Step 3: Stop words
What is stop words?
As the amount of data increases, so does the need for text mining, such as automatic text summarization. Hence, stop words removal is commonly used to reduce the size of the text and therefore increase the text mining speed and performance.
How it is performed?
In pre-processing, we have our own defined dictionary which comprises of stop words. This dictionary will then compare the given input text with its contents and remove the words which match.

FEATURE EXTRACTION

A. Word Frequency (F1)

Generally most frequent words in text are indicators of information. Relative word frequency of a word is defined as" the ratio of the number of occurrence of each word in the text over document length".

B. Length of the Sentence (F2)

It is applied to avoid the selection of too short and too long sentences. It is defined as "the ratio of word count in the sentence Si over word count in the longest sentence of the document ". Sentence length score is calculated as:

F2(Si)= NWS/NWLS

Where,

NWS= Number of words in sentence

NWLS= Number of words in longest sentence

C. Position of the Sentence (F3)

The positional value of a text entity can affect the summarization process. The text entity may be a word or sentence or phrase or paragraph. It is used by. The position score of a sentence can be found as: the first sentence in a heading has a score value of 5/5, the second sentence has a score 4/5, and so on.

D. Title Similarity (F4)

It is the word overlap between the sentence Si and the document title. It is calculated as:

F4(Si)= NTWS/NWT

Where,

NTWS= Number of title words in the sentence

NWT= Number of words in the title

E. Heading Similarity (F5)

It is the word overlap between the sentence Si and the document heading. It is calculated as follows:

F5 (Si)= NHWS/NWH

Where,

NHWS= Number of heading words in sentence

NWH= Number of words in heading

F. Sentence-to-sentence cohesion (F6)

It is defined as "the ratio of sum of similarity value of a sentence I with all other sentences over the largest raw feature value among all sentences in the document"

F6 (Si)= SSS/Max (SSS)

Where,

SSS= Sum of the sentence similarity.

Max (SSS) = Maximum value of sum of sentence similarity of document sentences.

SENTENCE SCORING AND RANKING

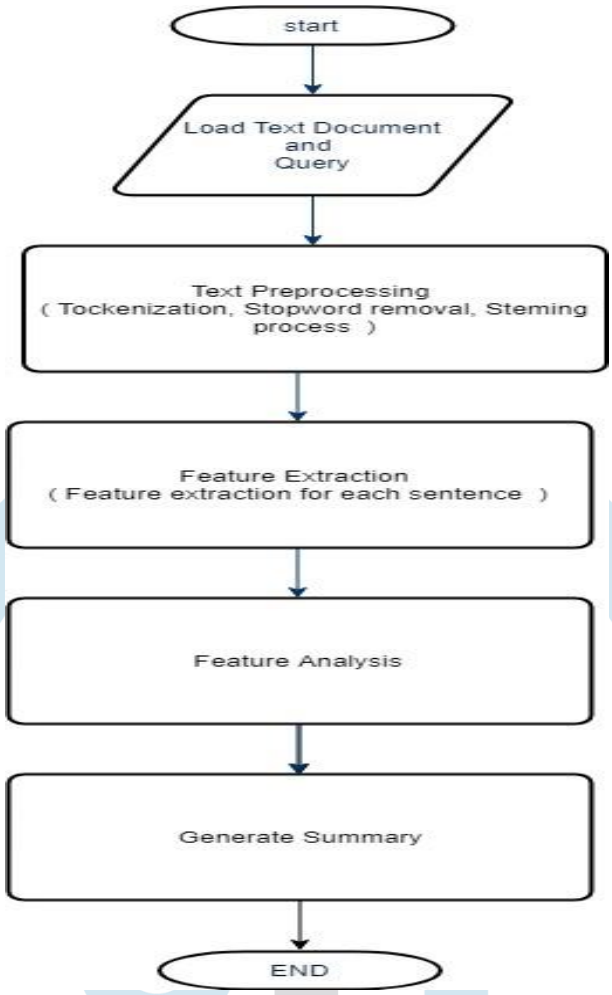
Calculate sentence score (Score(Si) = F1(Si)+F2(Si)+F3(Si)+F4(Si)+F5(Si)+F6(Si) /6)
and rank the sentences. Sentence scores are obtained by adding the feature scores.

SENTENCE EXTRACTION

Finally, select the top n sentences as the summary sentences, where the value of n depends on the compression rate We let the user choose how much summarization is to be done. We do that by taking amount of summary as an input.

MAIN SUMMARY

Order the summary sentences according to the order of the sentences in the original text and display as output summary text.



System Flow

IV. SCREENSHOT

V. CONCLUSION

An automatic extractive text summarizer should produce the summary quickly with no redundancy or minimum redundancy. Two kinds of techniques can be used for summary evaluation. They are intrinsic evaluation and extrinsic evaluation. Here we used intrinsic evaluation. In the proposed approach, we implement an automatic extractive text summarizer called heading and title wise summarizer. It provides feature based extractive single document heading or title wise summary of source document using statistical and linguistic features. The heading or title wise summarizer performs better than main summarizer and the remaining three summarizers used for evaluation. This approach improves significantly the coherence of the summary text by manually taking sentences from each heading. The coherent summary also improves the evaluation result. The performance of heading or title wise summarizer over main summarizer increases with the increase in summary length and the number of headings in the document. Our approach of heading wise summary generation for single document summarization could be applied for multi document summarization. Heading wise summarization that is multi heading summarization is a step toward multi document summarization.

VI. FUTURE SCOPE

We further focus on a multi document summarization. It is an automatic procedure aimed at extraction of information from multiple text written about same topic. The resulting summary report allow multiple users, such as professional information customers, to quickly familiarize themselves with information contained in a large cluster of document. In such a way, multi document summarization system are complementing the news aggregators performing the next step down the road of coping with information overload.

REFERENCES

[1] P.Krishnaveni, Dr.S. R. Balasundaram "Automatic Text Summarization by Local Scoring and Ranking for Improving Coherence" Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC)
[2] Reeta Rani and Sawal Tandon "LITERATURE REVIEW ON AUTOMATIC TEXT SUMMARIZATION" International Journal of Current Advanced Research ISSN: O: 2319-6475, ISSN: P:2319-6505
[3] Nedunchelian Ramanujam and Manivannan Kaliappan "An Automatic Multidocument Text Summarization Approach Based on Naive Bayesian Classifier Using Timestamp Strategy" Hindawi Publishing Corporation Scientific World Journal
[4] Mahsa Afsharizadeh, Hossein Ebrahimpour-Komleh, Ayoub Bagheri "Query-oriented Text Summarization using Sentence Extraction Technique" 2018 4th International Conference on Web Research (ICWR)
[5] ZHANG Pei-ying, LI Cun-he "Automatic text summarization based on sentences clustering and extraction" ©2009 IEEE
[6] Taeho Jo "K Nearest Neighbour for Text Summarization using Feature Similarity" 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan
[7] Mohamad Abdolahi, Shohreh Rad Rahimi, Ali Toofanzadeh Mozhdehi "An Overview on Extractive Text Summarization" 2017 IEEE 4th International Conference on Knowledge- Based Engineering and Innovation (KBEI) Dec. 22nd, 2017
[8] Vipul Dalal, Dr. Latesh Malik "A Survey of Extractive and Abstractive Automatic Text Summarization Technique" 2013 6th International Conference on Emerging Trends in Engineering and Technology.