Extractive Text Summarization Using Sentence Ranking

J.N.Madhuri
Dept. of Computer science and Engineering
CHRIST (Deemed to be University)
Bangalore, India
jn.madhuri@mtech.christuniversity.in

Ganesh Kumar.R,
Associate professor,
Dept. of Computer science and Engineering
CHRIST (Deemed to be University)
Bangalore, India
ganesh.kumar@christuniversity.in

***Abstract*— Automatic Text summarization is the technique to identify the most useful and necessary information in a text. It has two approaches 1) Abstractive text summarization and 2) Extractive text summarization. An extractive text summarization means an important information or sentence are extracted from the given text file or original document. In this paper, a novel statistical method to perform an extractive text summarization on single document is demonstrated. The method extraction of sentences, which gives the idea of the input text in a short form, is presented. Sentences are ranked by assigning weights and they are ranked based on their weights. Highly ranked sentences are extracted from the input document so it extracts important sentences which directs to a high-quality summary of the input document and store summary as audio.**

Keywords — Automatic text summarization, extractive summarization, sentence extraction, term weight, abstractive text summarization.

I. Introduction

In this present era, where huge quantity of information is generating on the internet day by day. So it is necessary to provide the better mechanism to extract the useful information fast and most effectively. Text summarization is one of the methods of identifying the important meaningful information in a document or set related document and compressing them into a shorter version preserving its overall meanings. It reduces the time required for reading whole document and also it space problem that is needed for storing large amount of data. Automatic text summarization problem has two sub-problems that is single document and multiple documents. In single document the single document is taken as the input and summarized information is extracted from that particular single document. In Multiple document the multiple documents of single topic is taken as an input and the output which is generated should be related to that topic.

In Automatic Text summarization has two approaches 1) Abstractive text summarization and 2) Extractive text summarization. An extractive text summarization means an important information or sentence are extracted from the given text file or original document. An extractive text summarization approach uses linguistic or statistical features for selecting useful informative sentence. An Abstractive text summarization will try to understand the input file or original file and re-generate the output in few words by identifying the main concept of the input file. In many research papers they have mentioned that extractive text summarization is sentence ranking. Extractive text summarization is divided in two phases: 1) Pre-processing 2) Processing. In this paper we are explaining extractive text summarization on single document.

II. Related work

This section describes about the methods that has been used for text summarization. In Natural language processing the text summarization is one of the fields. The text summarization is divided into 2 types: 1) Extractive summarization 2) Abstractive summarization. An Extractive text summarization is choosing necessary sentence from the text. This important paragraph can be selected by using linguistic and statistical features of paragraphs.

Abstractive summarization [6][7] understands the main concept and meaning of the given document or text. It finds the new concept from the document by using linguistic method by interpreting the text. The output which it generates will contain latest shorter version of text that contain important information of the document.

In earlier researches, Summarization was done on scientific documents based on the proposed features like phrase frequency, word (Luhn, 1958) [4], key phrases (Edmundson, 1969) [8] and position in the text (Baxendale, 1958) [5].

In 1958, most of the earlier works are done on the single document mainly focusing on technical document. (Luhn, 1958) [4], he has done his research on the extractive summarization. In his research he extracted important sentence by calculating word frequency and phrase frequency that gives the useful measure of its significance.

In 1958, Baxendale has done his research at IBM on Extractive summarization. He extracted important sentence by using the position of text. The author has tested 200 paragraphs towards his goal to find that in 85% of the sentences which author has taken first topic which is main topic sentence and the last sentence came 7%. The most accurate sentence would be selected from these two sentences.

In 1969, Edmundson has done research on extracted summarization in this he extracted important sentence by using two features position and word frequency importance were taken from the previous works. The author has added two they are: presence of cue words, and the skeleton of the document.

978-1-5386-9319-3/19/$31.00 ©2019 IEEE

III. APPROACH

In this proposed approach, we are using extractive method to get summary of given input. We are taking input as text file .txt.

1) Firstly, the file which is given as input is tokenized in order to get tokens of the terms

2) The stop words are removed from the text after tokenization. The words which are remained are considered as a key word

3) The key words are taken as an input for that we are attaching a part of tag to each key word

4) After completing this pre-processing step we are calculating frequency of each keyword like how frequently that key word has occurred from this maximum frequency of the keyword is taken.

5) Now weighted frequency of the word is calculated by dividing frequency of the keywords by maximum frequency of the key words

6) In this step we are calculating the sum of weighted frequencies.

Finally, summarizer will extract the high weighted frequency sentences and the extracted sentences are converted into audio form.

IV. METHODOLOGY

In the extractive summarization, the summarizer takes input as text file and tokenization of an input text is done in-order to remove find the terms of the text. Then stop words are removed in order to filter the text. And finally, part-of-speech tag is added to each token.

Step 1: After adding the parts-of-speech tag to tokens or terms each individual weight are assigned to the tokens. The term weight is calculated as follows:

$$\text{Wt} = \frac{\text{frequencyofterm}}{\text{Totalno.oftermsindocument}} \qquad (1)$$

Step 2: Now maximum weight of the token is considered after finding maximum weight. The weighted frequency of the document is calculated as follows:

$$\text{Wtf} = \frac{\text{frequencyofaterm}}{\text{maximumfrequencyoftheterm}} \qquad (2)$$

Step 3: In this step, the frequencies are connecting in place of corresponding words in sentence and sum of it is found. The ranks are found based on the weighted frequency. The sentences are sorted based on their Weighted frequency ranks like highest rank to lowest. The sentences are arranged in descending order.

Step 4: Finally, summarizer will extract sentences which rank is highest form the document and the sentences which are extracted are converted into audio (mp3) format.

V. ALGORITHM

Input: A text format of the data is taken as input.

Output: An appropriate summarized output text is generated which is shorter when compared to original. This extracted summarized output is converted into an audio form.

1. Reading the given text and the given text is tokenized
2. The stop-words are removed from the sentences.
3. Parts-of-speech are assigned to each token.
4. Weights are assigned to individual tokens by using formulae:

$$\text{Wt} = \frac{\text{frequencyofterm}}{\text{Totalno.oftermsindocument}} \qquad (1)$$

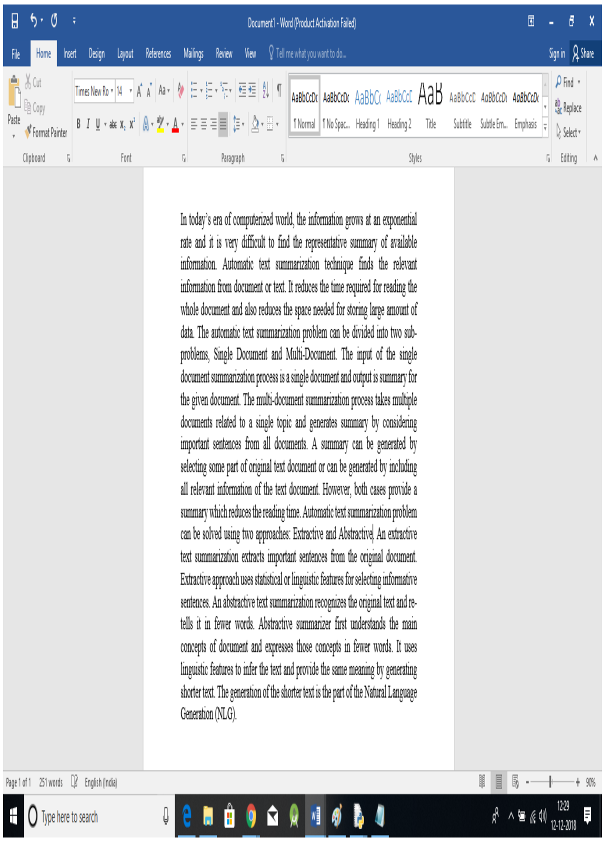
5. Weight frequency of tokens is calculated by using formulae:

$$\text{Wtf} = \frac{\text{frequencyofaterm}}{\text{maximumfrequencyoftheterm}} \qquad (2)$$

6. Individual terms ranks are calculated.
7. Finally, summarizer will extract the high weighted frequency sentences in order to find summary of a document and the extracted summaries are converted into audio form.

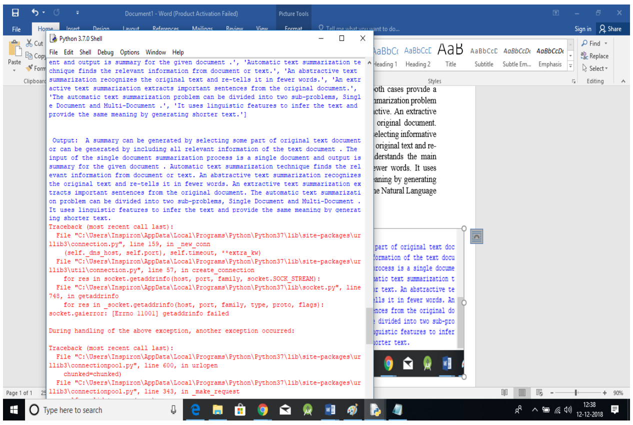
VI. RESULTS

The system with 5 documents is tested in the work, which contains 20 sentences. The sentences whose rank is greater than 8 are generated as an output by the summarizer. The summarized text is converted to an audio form. We have used Python 3.6 and NLTK to implement Extractive summarization and the input of the document is given in Figure1 and output of the document is in Figure2



In today's era of computerized world, the information grows at an exponential rate and it is very difficult to find the representative summary of available information. Automatic text summarization technique finds the relevant information from document or text. It reduces the time required for reading the whole document and also reduces the space needed for storing large amount of data. The automatic text summarization problem can be divided into two sub-problems, Single Document and Multi-Document. The input of the single document summarization process is a single document and output is summary for the given document. The multi-document summarization process takes multiple documents related to a single topic and generates summary by considering important sentences from all documents. A summary can be generated by selecting some part of original text document or can be generated by including all relevant information of the text document. However, both cases provide a summary which reduces the reading time. Automatic text summarization problem can be solved using two approaches: Extractive and Abstractive An extractive text summarization extracts important sentences from the original document. Extractive approach uses statistical or linguistic features for selecting informative sentences. An abstractive text summarization recognizes the original text and re-tells it in fewer words. Abstractive summarizer first understands the main concepts of document and expresses those concepts in fewer words. It uses linguistic features to infer the text and provide the same meaning by generating shorter text. The generation of the shorter text is the part of the Natural Language Generation (NLG).

Fig. 1. Input text



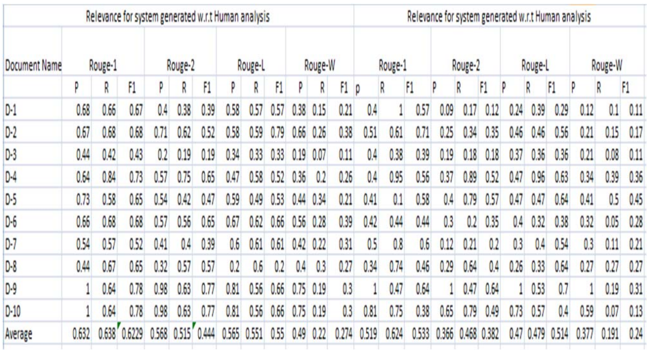
```
ent and output is summary for the given document .', 'Automatic text summarization te
chnique finds the relevant information from document or text.', 'An abstractive text
summarization recognizes the original text and re-tells it in fewer words.', 'An extr
active text summarization extracts important sentences from the original document.',
'The automatic text summarization problem can be divided into two sub-problems, Singl
e Document and Multi-Document .', 'It uses linguistic features to infer the text and
provide the same meaning by generating shorter text.']


 Output:  A summary can be generated by selecting some part of original text document
or can be generated by including all relevant information of the text document . The
input of the single document summarization process is a single document and output is
summary for the given document . Automatic text summarization technique finds the rel
evant information from document or text. An abstractive text summarization recognizes
the original text and re-tells it in fewer words. An extractive text summarization ex
tracts important sentences from the original document. The automatic text summarizati
on problem can be divided into two sub-problems, Single Document and Multi-Document .
It uses linguistic features to infer the text and provide the same meaning by generat
ing shorter text.
Traceback (most recent call last):
  File "C:\Users\Inspiron\AppData\Local\Programs\Python\Python37\lib\site-packages\ur
llib3\connection.py", line 159, in _new_conn
    (self._dns_host, self.port), self.timeout, **extra_kw)
  File "C:\Users\Inspiron\AppData\Local\Programs\Python\Python37\lib\site-packages\ur
llib3\util\connection.py", line 57, in create_connection
    for res in socket.getaddrinfo(host, port, family, socket.SOCK_STREAM):
  File "C:\Users\Inspiron\AppData\Local\Programs\Python\Python37\lib\socket.py", line
748, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
socket.gaierror: [Errno 11001] getaddrinfo failed

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\Inspiron\AppData\Local\Programs\Python\Python37\lib\site-packages\ur
llib3\connectionpool.py", line 600, in urlopen
    chunked=chunked)
  File "C:\Users\Inspiron\AppData\Local\Programs\Python\Python37\lib\site-packages\ur
llib3\connectionpool.py", line 343, in _make_request
```

Fig. 2. Output generated by system



Relevance for system generated w.r.t Human analysis													Relevance for system generated w.r.t Human analysis											
Document Name	Rouge-1			Rouge-2			Rouge-L			Rouge-W			Rouge-1			Rouge-2			Rouge-L			Rouge-W		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	p	R	F1	P	R	F1	P	R	F1	P	R	F1
D-1	0.68	0.66	0.67	0.4	0.38	0.39	0.58	0.57	0.57	0.38	0.15	0.21	0.4	1	0.57	0.09	0.17	0.12	0.24	0.39	0.29	0.12	0.1	0.11
D-2	0.67	0.68	0.68	0.71	0.62	0.52	0.58	0.59	0.79	0.66	0.26	0.38	0.51	0.61	0.71	0.25	0.34	0.35	0.46	0.46	0.56	0.21	0.15	0.17
D-3	0.44	0.42	0.43	0.2	0.19	0.19	0.34	0.33	0.33	0.19	0.07	0.11	0.4	0.38	0.39	0.19	0.18	0.18	0.37	0.36	0.36	0.21	0.08	0.11
D-4	0.64	0.84	0.73	0.57	0.75	0.65	0.47	0.58	0.52	0.36	0.2	0.26	0.4	0.95	0.56	0.37	0.89	0.52	0.47	0.96	0.63	0.34	0.39	0.36
D-5	0.73	0.58	0.65	0.54	0.42	0.47	0.59	0.49	0.53	0.44	0.34	0.21	0.41	0.1	0.58	0.4	0.79	0.57	0.47	0.47	0.64	0.41	0.5	0.45
D-6	0.66	0.68	0.68	0.57	0.56	0.65	0.67	0.62	0.66	0.56	0.28	0.39	0.42	0.44	0.44	0.3	0.2	0.35	0.4	0.32	0.38	0.32	0.05	0.28
D-7	0.54	0.57	0.52	0.41	0.4	0.39	0.6	0.61	0.61	0.42	0.22	0.31	0.5	0.8	0.6	0.12	0.21	0.2	0.3	0.4	0.54	0.3	0.11	0.21
D-8	0.44	0.67	0.65	0.32	0.57	0.57	0.2	0.6	0.2	0.4	0.3	0.27	0.34	0.74	0.46	0.29	0.64	0.4	0.26	0.33	0.64	0.27	0.27	0.27
D-9	1	0.64	0.78	0.98	0.63	0.77	0.81	0.56	0.66	0.75	0.19	0.3	1	0.47	0.64	1	0.47	0.64	1	0.53	0.7	1	0.19	0.31
D-10	1	0.64	0.78	0.98	0.63	0.77	0.81	0.56	0.66	0.75	0.19	0.3	0.81	0.75	0.38	0.65	0.79	0.49	0.73	0.57	0.4	0.59	0.07	0.13
Average	0.632	0.638	0.6229	0.568	0.515	0.444	0.565	0.551	0.55	0.49	0.22	0.274	0.519	0.624	0.533	0.366	0.468	0.382	0.47	0.479	0.514	0.377	0.191	0.24

Fig. 3. Performance comparison between results obtained by system generated w.r.t Human Analysis and Ms word generated w.r.t Human Analysis

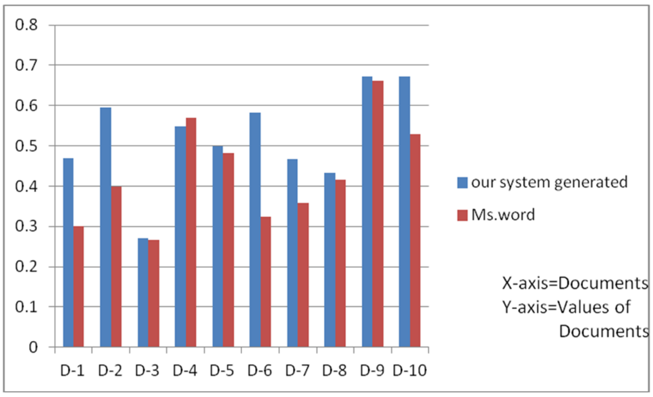


Fig. 4. Comparison between MS Word and system generated

An evaluation of the proposed system is done with MSWord and human summarized data generated by system is shown in the above figure (i.e. Figure: 3). system generated summary is also calculated with human generated summary. The result is specified in the above figure (Figure: 3) and the pictorial illustration of the both the system relevancy with respect to human summarized data is shown in the above figure (Figure: 4).

VII. CONCLUSION

Automatic text summarization is a complex task which contains many sub-tasks in it. Every subtask has an ability to get good quality summaries.

The important part in extractive text summarization is identifying necessary paragraphs from the given document. In this work we proposed extractive based text summarization by using statistical novel approach based on the sentences ranking the sentences are selected by the summarizer. The sentences which are extracted are produced as a summarized text and it is converted into audio form. The proposed model improves the accuracy when compared traditional approach.

REFERENCES

[1] Nenkova, A.(2011).“Automatic summarization, Foundations and Trends in Information Retrieval”,5(2),103-233

[2] Gupta,V and Lehal,G.s (2010). “A survey of text summarization extractive techniques.” Journal of Emerging Technologies in Web Intelligence,2(3),258-268

[3] Goldstein.J,carbonell.J,Kantrowitzt.M(1998).“Multiple-document summarization by sentence Extraction”40-48

[4] Weigo Fan, Linda Wallace, Stephanie Rich and Zhongju Zang,: “Tapping the power of text mining”, Journal of ACM,Blacksburg 2005.

[5] Baxendale,P.(1958). “Machine-made index for technical literature” –an experiment. IBM Journal of Research developement354-361

[6] Vishal Gupta,G.s. Lehal. “A survey of text mining techniques and applications”, Journal of Emerging Technologies in Web intelligence,VOL 1,NO 1,6076,August 2009

[7] G.Erkan, Dragomir R.Radev. “LexRank:graph based centrality as salience in Text summarization”,Journal of Artificial intelligence Research,Re-search,vol.22,pp.457-479 2004

[8] Luhn, H (1958). “The automatic creation of literature abstracts”. IBM Journal of Research Development, 2(2):159-165.

[9] StanfordCoreNLP,From http://nlp.stanford.edu/software/corenlp.shtml

[10] Apache Open NLP. From http://opennlp.apache.org/

[11] Natural Language Toolkit. From http://nltk.org/

[12] Rapid Miner. From Available: http://rapidminer.com/