International Journal of Scientific Research and Engineering Development-– Volume2 Issue 6, Nov-Dec 2019
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Text Summarizer using NLP

Pooja Mudgil*, Akshit Gupta**, Mohammed Ayaan Abbasi***, Prashant Verma****, Vipul Anand*****
*(*Assistant Professor, Computer Science Department, Bhagwan Parshuram Institute of Technology, Delhi, India*
Email: engineer.pooja90@gmail.com)
** (*Student, Information Technology Department, Bhagwan Parshuram Institute of Technology, Delhi, India*
Email: akshit.gupta98@gmail.com)
*** (*Student, Information Technology Department, Bhagwan Parshuram Institute of Technology, Delhi, India*
Email: ayaan.abbasi0@gmail.com)
**** (*Student, Information Technology Department, Bhagwan Parshuram Institute of Technology, Delhi, India*
Email : prashantverma.vn@gmail.com)
***** (*Student, Information Technology Department, Bhagwan Parshuram Institute of Technology, Delhi, India*
Email: vipulanand1498@gmail.com)

----------**********----------

Abstract:

The analysis and processing of large amount of data is a main field of interest for researchers and developers. Text summarization is one of the sub fields of data and text analysis. Text summarization is a common problem in machine learning and natural language processing (NLP). There is a need to develop some machine learning algorithms that can automatically shorten the text and provide accurate and precise summary that can be easily utilized. The purpose of text summarization is to create a meaningful, logical and fluent summary having only key points mentioned in a particular document. This technique reduces reading time and researching time for information.

----------**********----------

I. INTRODUCTION

Natural Language Processing (NLP) is a subfield of computer science, information engineering and artificial intelligence that deals with the interactions between computers and human languages[1]. It is broadly defined as automatic manipulation of natural language, like speech and text. NLP also conducts information extraction and retrieval, sentiment analysis and more. Some well-known application areas of NLP are Optical Character Recognition (OCR), Speech Recognition, Machine Translation and Chabot[2].

Text Summarization is one of those applications of Natural Language Processing (NLP) which is used to generate a concise and meaningful summary of text from multiple text resources such as news articles, blog posts, research papers, etc[3]. The demand for automatic text summarization systems is being forth these days due to the availability of large amounts of textual data.

Text Summarization can be broadly divided into two categories: -

1. Extractive Summarization: These methods involve fetching key points form source documents and combining them together to make a summary. Hence, identifying the right sentences for summarization is of utmost importance in an extractive method.[4]

2. Abstractive Summarization: These methods use advanced NLP techniques to generate an entirely new summary with new phrases and

sentences that includes the most useful information from the original text.[5]

II. RELATED WORK

Hongyan Jing and Kathleen R. Mckeown in their paper have proposed "cut and paste" technique[6] to summarize the text. They shorten the sentence by cutting and pasting important words only. They have proposed the cut and paste architecture as according to them, this technique will result in human written abstract.

In this paper, they[7] have proposed multi-document summarization technique using A* search. Multi-document summarization aims to produce a short summary of multiple document. They have focused solely on extractive type summarization. Their research challenges the search problem of calculating and printing best scoring summary for a given set of documents.

Using Lexical Chains for Text Summarization[8]. In this paper, they have proposed a way to shorten a source text preserving its information content. It can give an analysis of a scientific field quickly generating indicative field on the general topic of a text. It indicates whether a text is worth reading or not. They have enquired a method for the production of such indicative summaries from arbitrary text. The summary generation involves three types of source text: linguistic, domain & communicative. This type of summary generation is very efficient and but it is domain dependent.

Salience Based Content Characterization of Text documents[9]. They have implemented an approach for content characterization of text documents. It is independent of domain and genre. Hence, it doesn't require in-depth analysis of a text document for full meaning. In this, phrasal expressions are used rather than sentences or paragraphs which can be characterized as a salience based content. Basically, it is an appropriate approach to know what the document is about.

In our research and implementation, we have developed an automatic text summarization technique using the extractive algorithm. We calculated the frequency score of each sentence on the basis of frequency score of each word in the respective sentence. The sentence whose frequency score is greater than the average frequency score of all sentences is included in the summary.

III. PROPOSED ARCHITECTURE

The existing algorithms used for text summarization includes abstraction and extraction algorithms. The extraction based algorithm works by extracting key phrases from the given text and combining them to make a meaningful summary. The key phrases are defined as the most frequent words in the text or the words which defines the meaning of the sentence.

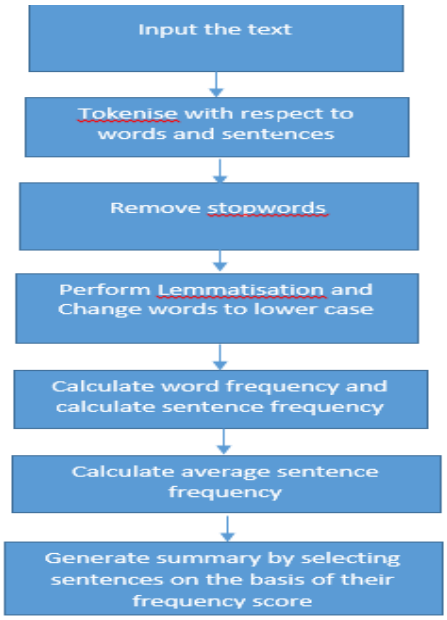

Fig-1 Proposed Architecture

The abstraction based algorithm emphasizes on paraphrasing and shortening of source document. When abstraction based algorithm is applied on deep learning problems, it can overcome the precision and accuracy problem in grammar inconsistencies faced by extractive method. The abstractive based algorithm works by creating new sentences and phrases which are dependent on the original text or source document.

The proposed algorithm for text summarization is based on extractive algorithm[10]. The algorithm works by performing tokenization on the input text with respect to words. After tokenization is done, all words are changed to lower case and stop words are removed from the text. After stop words removal, lemmatization is performed on the text. Now, frequency of words is calculated in each sentence and stored in a list. Average is

calculated by dividing the frequency score of each sentence by length of the list. If the frequency score of sentence is higher than average, it is included in the summary else the sentence is discarded.

The detailed working of each task performed during the process is described below –

1. Tokenization – tokenization is defined as breaking of text into sentences and words. It is of two types mainly sentence tokenization and words tokenization. In sentence tokenization, the input text is broken down into sentences. In word tokenization, the input text is broken down into words. We have implemented word tokenization is our algorithm.

2. Lower case – It is defined as the process of modifying all the tokenized words into lower case.

3. Stop words removal – stop words are defined as the words which occur frequently in most of the sentences and add little or no meaning to the definition of the sentence. Stop words are removed before summarizing the text.

4. Lemmatization – lemmatization and stemming are defined as the process of reducing a word to its simpler form. Both of the techniques perform the same function. The main difference between lemmatization and stemming is the accuracy. Lemmatization is more successful in term of accuracy and precision as compared to stemming. We have implemented lemmatization in our algorithm.

5. Frequency calculation – the frequency of words in a sentence is calculated and a frequency score is set for each sentence based on the frequency of words.

6. Average score calculation – average score is calculated by dividing the sum of all the sentence frequencies by number of frequencies.

7. Sentence selection – the summary is generated on the basis of sentence selection. If the frequency score of sentence is greater than the average score, it is included in the summary else it is discarded from the summary.

IV. RESULT

Following steps depict the workflow of the extraction based algorithm implemented for summary generation:

a) Text whose summary is to be generated is taken as an input from the user with the help of an interactive user interface.

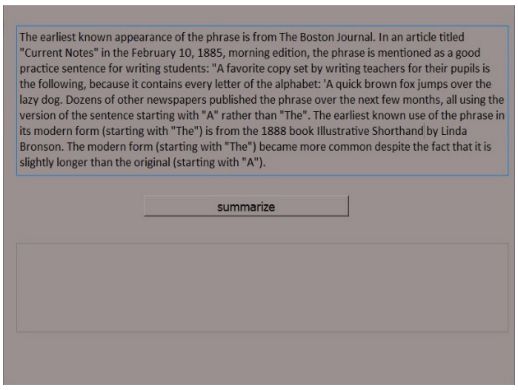

Fig-2 Input from user

b) When the summarize button is pressed the text from the input is converted to plaintext and is tokenized into words and sentences followed by stop words removal.

```
def generate_summary(self):
    text = self.textEdit_3.toPlainText()
    words = word_tokenize(text)
    sentences = sent_tokenize(text)
    sWords = set(stopwords.words("english"))
    w_net = WordNetLemmatizer()
```

Fig-3 Removal of stop words

c) During the third step all the words in the sentence are lemmatized and frequency of each word is calculated followed by frequency calculation of the said word in the sentences and a dictionary is maintained containing all the frequency values.


```
freq_dist = dict()

for word in words:
    word = word.lower()

    if word in sWords:
        continue

    word = w_net.lemmatize(word, pos='v')

    if word in freq_dist:
        freq_dist[word] += 1
    else:
        freq_dist[word] = 1

sent_dist = dict()

for sentence in sentences:
    for word, freq in freq_dist.items():
        if word in sentence:
            if sentence in sent_dist:
                sent_dist[sentence] += freq
            else:
                sent_dist[sentence] = freq
```

Fig-4 Frequency calculation

d) During the final step of the algorithm average is calculated based on the dictionary values and a summary is created with the help of this average value.

```
avg = int(sum(sent_dist.values()) / len(sent_dist))
summary = ""
for sentence in sentences:
    if sent_dist[sentence] > avg * 1.1:
        summary += " " + sentence

self.textEdit_4.setText(summary)
```

Fig-5 Calculation of average

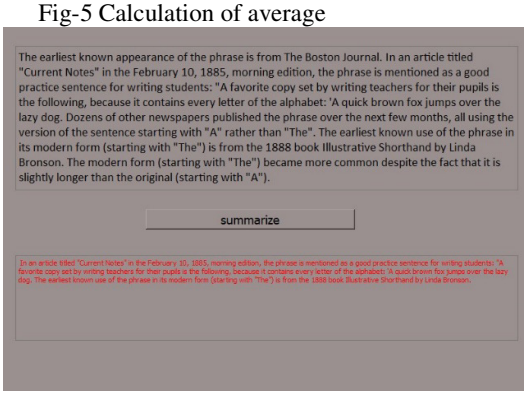

Fig-5 Summarized text

Figure-5 displays the summary generated (red in color) with respect to the corresponding input text entered by the user.

V. CONCLUSION

In this paper, we have implemented Text Summarization for converting large textual information into a short, brief and crisp summary by using an extractive algorithm. The algorithm works by performing tokenization on the input text with respect to words. After tokenization is done, all words are changed to lower case and stop words are removed from the text. After stop words removal, lemmatization is performed on the text. Now, frequency of words is calculated in each sentence and stored in a list. Average is calculated by dividing the frequency score of each sentence by length of the list. If the frequency score of sentence is higher than average, it is included in the summary else the sentence is discarded. Then the resultant summary is printed, giving us the accurate gist of the original text.

VI. REFERENCES

[1] A Survey of Text Summarization Extractive Techniques Vishal Gupta University Institute of Engineering & Technology, Computer Science & Engineering, Panjab University Chandigarh, India, Email: vishal@pu.ac.in Gurpreet Singh Lehal Department of Computer Science, Punjabi University Patiala, Punjab, India, Email: gslehal@yahoo.com

[2] E. H. Hovy, Automated Text Summarization. The Oxford Handbook of Computational Linguistics, Chapter 32, pages 583-598. Oxford University Press, 2005.

[3] F. C. Pembe and T. Güngör, "Automated Query-biased and Structure-preserving Text Summarization on Web Documents," in Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul, June 2007.

[4] G. Yihong, X. Liu. "Generic text summarization using relevance measure and latent semantic analysis." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2001.

[5] S. Alfayoumy, J. Thoppil, A Survey of Unstructured Text Summarization Techniques, International Journal of Advanced Computer Science and Applications, Vol. 5, No. 4, 2014.

[6] Hongyan Jing and Kathleen R. McKeown Department of Computer Science Columbia University New York, NY 10027, USA hjing, kathyQcs.columbia.edu.

[7] Multi-document summarization using A* search and discriminative training Ahmet Aker

Trevor Cohn Department of Computer Science University of Sheffield, Sheffield, S1 4DP, UK.

[8] Using Lexical Chains for Text Summarization. Regina Barzilay and Michael Elhadad Mathematics Computer Science Dept. Ben Gurion University in the Negev Beer-Sheva, 84105 Israel.

[9] Salience Based Content Characterization of Text Documents. Branimir Bougurev Watson Research Center, Yorkton Heights,NY10598bkbwatson@ibm.com and Christopher Kennedy, Department of Linguistics, Northwestern University, 2016 Sheridan Road, Evanston 60208 kennedy@ling.nwu.edu.

[10] L. Suanmali, N. Salim, M. S. Binwahlan, "Fuzzy Logic Based Method for Improving Text Summarization", International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009.

