Original Article

Cluster Based Text Summarization Using Cosine Similarity

Aishwarya Bindal, Anshika Rajput, Abhiraj, Atul Sharma

Department of Computer Science and engineering.
Meerut Institute of Engineering and Technology, Meerut, U.P., India
{aishwaryabindal2001@gmail.com, anshikarajput171@gmail.com, bhadanaabhirajcr7@gmail.com, atulsharma2823@gmail.com }
DOI: 10.47750/pnr.2022.13.S10.246

Abstract

In this paper, the following Creation of short summaries, covering the main points of the given text, preserving the overall meaning of text, reduction in reading time, providing instant response, increasing the productivity level.
The proposed project "Cluster based text summarization using cosine similarity" is used for creating short and precise summaries that will cover all the main points of the given text, preserving the overall meaning of the text.

Keyword: Numpy, Unsupervised Learning (Clustering), Cosine Similarity Technique

1. Introduction

The process that provides short and accurate summary of long texts is known as text summarization. The main focus of this technique is to create summaries that provide the most beneficial information of the texts without changing the meaning of texts.

Earlier when there was no such facility, people tend to read the whole texts and then summarize the main points of that text. This results in consumption of immense time. Also, there are wide chances of missing some important points.

However with time, some techniques are invented to solve this problem. These are mainly categorized into two types that are extractive summarization and abstractive summarization.

Following research paper discuss about cluster based text summarization using cosine similarity technique for making short and precise summary of long texts.

A summary is a group of sentences that is the short description of a long text. It gives us brief about that long text.

Now talking about summarization of text, it is the method of creating summaries from given long texts with the help of various techniques and methods.

Text summarization is broadly classified in 2 categories**1. Extractive**

Summarization

In this type of summarization, essential sentences of the text are identified and are used in the summary. In this exact sentences for the original text are used in the generated summary.

2. Abstractive Summarization

In this type of summarization, sentences are not selected from the original text rather it identifies the important points of the text and then generate new phrases either by rephrasing or by using the words that are different from the word that are used in the original texts [6],[7],[8].

Though there are various techniques for text summarization, in this research paper we are focusing on Clustering technique for summarization process.

2. Techniques used for text summarization

There are various techniques and methods that are used for text summarization. In this section we have given brief of some of these techniques-

2.1 Luhn's Heuristic Method

In this the main approach is that the sentence which have the highest occurrence of the words that have the maximum frequency are considered important than others. Drawback: It doesn't provide much accurate results and is very old technique.

2.2Edmundsons's Heuristic Method

In this more importance is given to the words that are present in the title of the document. It is similar to Luhn's method.
Drawback: Less Accuracy and it is an old approach.

2.3KL-Sum

In this those sentences that minimize the Summary Vocabulary Divergence from the original Inout Vocabulary are included in the summary. Drawback: There is no explicit way to remove redundancy [9], [10],[11].

2.4TF IDF(Term Frequency Inverse Document Frequency) Method

It is an extractive approach for summarizing the text. It is broadly the multiplication of Term Frequency and IDF statistics [12].
TF stands for the term frequency. It depicts how many times the word has occurred in the file.IDF stands for Inverse Document Frequency. It depicts how much information the words provide. In simple terms it tells if the given word is either frequent or infrequent across every file [13].
In this, text is converted into sentences then it is reprocessed by removing stopwords, punctuation etc. After this TF matrix and IDF matrices are created. Then tf-idf values are calculated and with that sentence scores are calculated. Then threshold is determined to generate summary [14].

3. Design and Implementation of Text Summarization

This project includes the following modules-

a) Libraries-

The following libraries are includednltk, spacy, network, IPython, numpy, playsound. pip is used to install libraries such as pysoundfile, bitstring, gTTS.

b) Reading a file content-

In this a function is made to read the file content and then the sentences of the document are being splitted into different sentences when a fullstop(.) is reached.

Algorithm is as follows:

```
def read_document(file_name):
        file1 = open(file_name,'r')
file_data = file1.readlines()
article1 = file_data[0].split('.')
sentence = []
```

c) Stopwords-

Stopwords are the set of words that are commonly used in a language(example- a, the, is etc.). These words needs to be filtered out before processing the text because these words did not add much meaning to the text. In this project we are taking the top hundred stop words of english language [15], [16].

Algorithm is as follows:
nltk.download('stopwords') l = stopwords.words('english') l[0:100]

d) Finding sentence similarity-

In this module the similarity measure between two sentences is being calculated.
The algorithm is as follows:

```
def sentence_similarity_measure(sentence1, sentence2, stopwords = None): if stopwords is None:
stopwords = [] sentence1 = [t.lower() for t in sentence1]
sentence2 = [t.lower() for t in sentence2] total_words =
list(set(sentence1 + sentence2)) vector_1 = [0] *
len(total_words) vector_2 = [0] * len(total_words) for t
in sentence1: if t in stopwords:
continue vector_1[total_words.index(t)] +=
1 for t in sentence2: if t in stopwords:
continue
vector_2[total_words.index(t)] += 1 return 1 -
cosine_distance(vector_1, vector_2) For eg:
```

There are two sentences: This is a dog. This thing here refers to an animal called dog.
So the value of similarity measure between the two sentences is 0.7781047020641257.

e) Building the similarity matrix-

These are used for alignment of sequences. It tells the similarity between any two sentences [17], [18]. The value of similarity measure will be greater if the two objects have greater similarity. The algorithm is as follows:

```
def make_similarity_matrix(sentence,stop_words) similarity_matrix =
np.zeros((len(sentence), len(sentence))) print(similarity_matrix) for
index1 in range(len(sentence)): for index2 in range(len(sentence)): if
index1 == index2:
continue
similarity_matrix[index1][index2] = sentence_similarity_measure(sentences[index1], s entences[index2],
stop_word) return similarity_matrix
```

f) Generating the Summary-

In this module summary of the document is generated. Here top five sentences are being taken to generate the summary [19], [20]. This is a changeable number, we can set it according to our need. In this top ranked sentences are joined together according to their indexes to generate the summary. def make_summary(file_name, high_rank = 5):

```
stop_words = stopwords.words('english')
summarize_content = [] sentence =
read_article(file_name)
sentence_similarity_matrix = make_similarity_matrix(sentence, stop_words)
print(sentence_similarity_matrix)
sentence_similarity_graph = nx.from_numpy_array(sentence_similarity_matrix)
print(sentence_similarity_graph)
values = nx.pagerank(sentence_similarity_graph) print(values)
rank_sentence = sorted(((values[i],s) for i,s in enumerate(sentences)), reverse = True) print("Indexes of
highest ranked sentences are ", rank_sentence) for i in range(high_rank):
summarized_content.append(" ".join(ranked_sentence[i][1])) print("Summarized content is :
\n",". ".join(summarize_content))
```

g) Print the summary-

In this module, the final summary is being printed [21]. The name of the document whose summary we want to generate is given. Eg-generate_summary('/filename.txt')

h) Audio alert-

After summary generation ,an audio alert is played to show that the summary is generated successfully.

```
import IPython
IPython.display.Audio('/audiofile.m4a',autoplay=True)
```

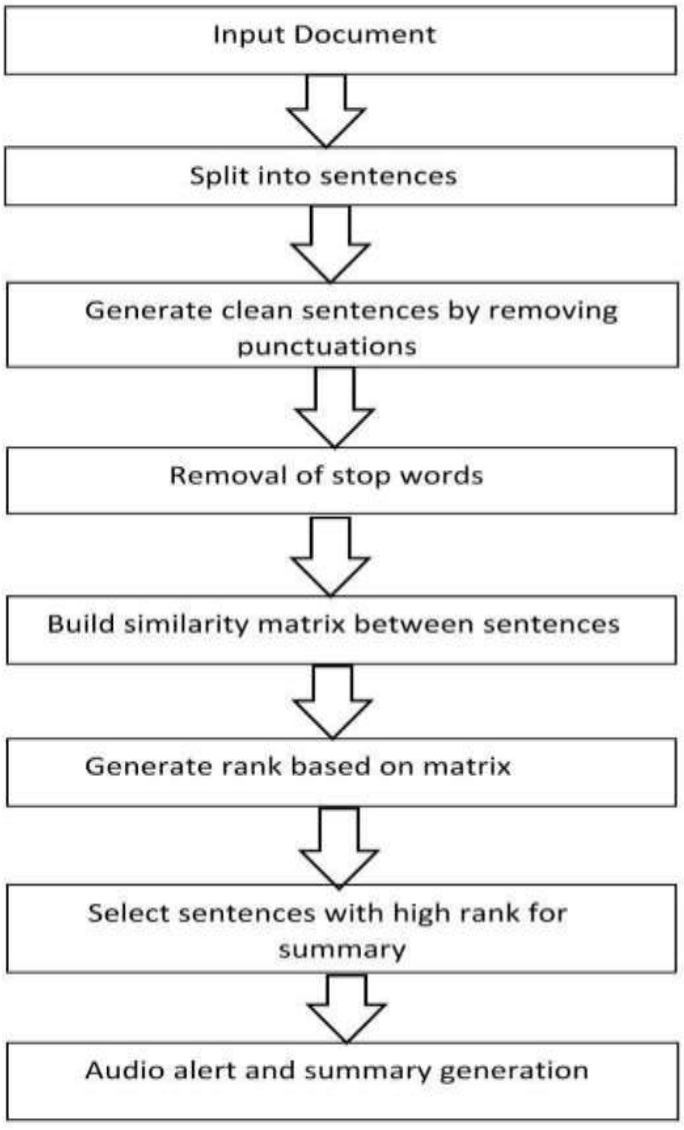

Fig 1. Work flow of the text Summarization

In fig 1, a document is taken as input. Initially this document is splitted into sentences. After this, preprocessing is done. Clean sentences are generated by removal of punctuation marks and stop words. Then similarity matrix is build. Rank is generated based on the matrix. Now, sentences with higher rank are selected for the summary. At last audio alert is sent indicating that summary is generated and summary is displayed.

4. Experimental Result Analysis

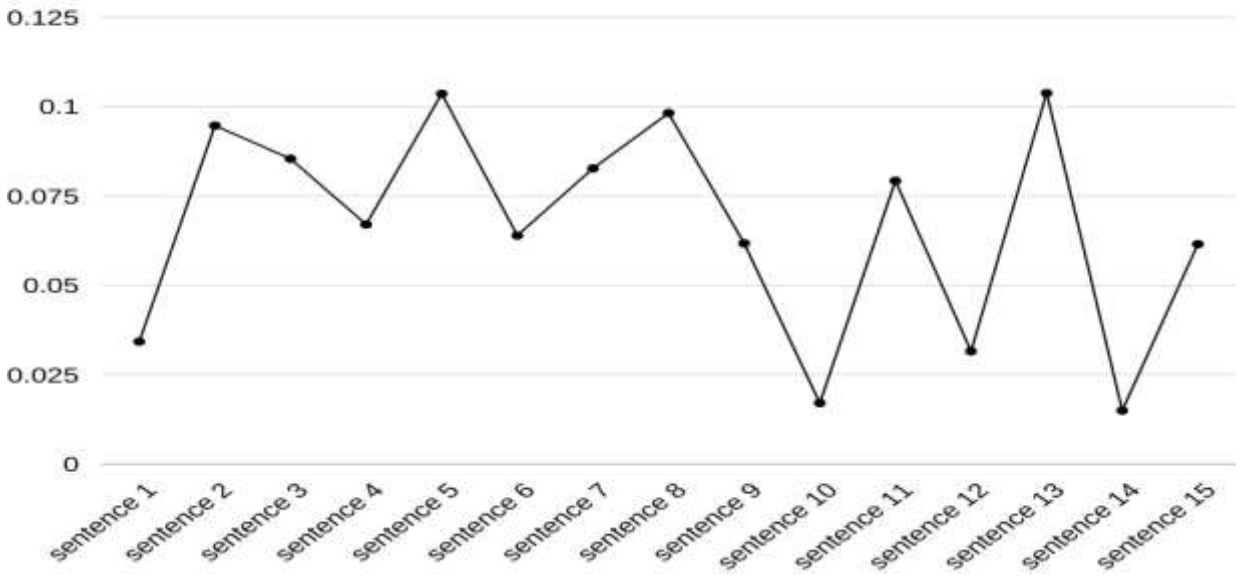

Fig 2. Graph showing average similarity measure of each sentence

Sentence Number	Similarity Measure
Sentence 1	0.0342
Sentence 2	0.0947
Sentence 3	0.0854
Sentence 4	0.0670
Sentence 5	0.1036
Sentence 6	0.0639
Sentence 7	0.0827
Sentence 8	0.0982
Sentence 9	0.0617
Sentence 10	0.0170
Sentence 11	0.0792
Sentence 12	0.0315
Sentence 13	0.1038
Sentence 14	0.0149
Sentence 15	0.0615

Fig 3. Table of sentence no. and their average similarity measure value

The input document consist of following paragraph-

Everything we are surrounded by, the trees, water, air, grass, furniture in our house, makes up the environment.

The environment makes it possible for us to live on earth, and it keeps a strict balance between the elements present on earth. However, lately, the balance has been disrupted.

Natural resources are diminishing,mainly due to the misuse and greed of human beings.

One of the basic requirements of the earth is keeping the environment clean.

Not only are we failing to maintain that, but the environment's condition is also becoming worse day by day.

The pollution caused by human beings has an adverse effect on the environment, and pollution harms the environment and affects us.

Specific measures need to be taken to curb the toxicity caused to the environment. Plastics and papers need to be recycled and reused, and more trees should be planted for purifying the air.

People should carpool together instead of traveling in private cars.

A fine should be issued for people who litter the streets with garbage.

If these measures are not taken, the earth will gradually become unfit for living.

A clean and proper environment is needed for all living creatures existence.

With the increase of environmental depletion, the government has also started taking things seriously.

However, it is not enough; without the people's co-operation, nature will continue on its path of degradation, and someday we will be wiped from existence.

The above paragraph of 15 sentences is taken as input. Each sentence is then compared with other sentences and sentence similarity is checked. The average of similarity measure of a particular sentence with all every sentence of the file is calculated.

The above table mentions the average similarity measure of each sentence. Since the average of sentence number 13 is maximum so it is provided with rank one. Sentence number 5 has second highest average so it is provided with rank 2. 8th sentence has 3rd highest value of average so rank 3 is allotted to it. Sentence 2 and sentence 3 are given 3th and 5th rank respectively.

Since the program is set to take top 5 sentences of the paragraph so sentences ranked from 1 to 5 are taken to form the summary.

So the generated summary will be-

A clean and proper environment is needed for all living creatures existence. One of the basic requirements of the earth is keeping the environment clean. Specific measures need to be taken to curb the toxicity caused to the environment. The environment makes it possible for us to live on earth, and it keeps a strict balance between the elements present on earth. However, lately, the balance has been disrupted.

5. Conclusion

The proposed project "Cluster based text summarization using cosine similarity" is used for creating short and precise summaries that will cover all the main points of the given text, preserving the overall meaning of the text.

This project deals with extractive text summarization. In this cluster based approach is used along with cosine similarity technique for summarizing the text.

This project generates a short and precise summary of given number of sentences along with an audio alert that the summary is generated successfully. As per the result analysis, we see the better accuracy in this project.

6. References

The following material were referenced in developing this research paper:

[1] "A Survey of Text SummarizationExtractive Techniques" by Vishal Gupta, Gurpreet Singh Lehal, vol. 2, no. 3,Journal of emerging technologies in web intelligence, Aug. 2010.

[2] "Multi-document summarization", Wikipedia, 2015.

[3] Kamal Sarkar, "Sentence Clustering-based Summarization of Multiple Text Documents", TECHNIA – International Journal of ComputingScience and Communication Technologies, vol. 2, no. 1, Jul. 2009.

[4] A. P. Siva kumar, Dr. P. Premchand and Dr. A. Govardhan, "Query Based Summarizer Based on Similarity of Sentences and Word Frequency", International Journal of Data Mining &KnowledgeManagement Process, vol.1, no.3, May 2011.

[5] "Cosine similarity", Wikipedia, the free encyclopedia, 2015.

[6] Narayan, Vipul, and A. K. Daniel. "Design consideration and issues in wireless sensor network deployment." (2020): 101-109.

[7] Choudhary, Shubham, et al. "Fuzzy approach-based stable energy-efficient AODV routing protocol in mobile ad hoc networks." Software Defined Networking for Ad Hoc Networks. Cham: Springer International Publishing, 2022. 125-139.

[8] Narayan, Vipul, and A. K. Daniel. "RBCHS: Region-based cluster head selection protocol in wireless sensor network." Proceedings of Integrated Intelligence Enable Networks and Computing: IIENC 2020. Springer Singapore, 2021.

[9] Narayan, Vipul, and A. K. Daniel. "CHOP: Maximum coverage optimization and resolve hole healing problem using sleep and wake-up technique for WSN." ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 11.2 (2022): 159-178.

[10] Narayan, Vipul, and A. K. Daniel. "CHHP: coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network." International Journal of System Assurance Engineering and Management 13.Suppl 1 (2022): 546-556.

[11] Narayan, Vipul, and A. K. Daniel. "IOT based sensor monitoring system for smart complex and shopping malls." Mobile Networks and Management: 11th EAI International Conference, MONAMI 2021, Virtual Event, October 27-29, 2021, Proceedings. Cham: Springer International Publishing, 2022.

[12] Narayan, Vipul, and A. K. Daniel. "Energy Efficient Protocol for Lifetime Prediction of Wireless Sensor Network using Multivariate Polynomial Regression Model." Journal of Scientific & Industrial Research 81.12 (2022): 1297-1309.

[13] Awasthi, Shashank, et al. "A Comparative Study of Various CAPTCHA Methods for Securing Web Pages." 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE, 2019.

[14] Narayan, Vipul, and A. K. Daniel. "FBCHS: Fuzzy Based Cluster Head Selection Protocol to Enhance Network Lifetime of WSN." ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 11.3 (2022): 285-307.

[15] Narayan, Vipul, et al. "E-Commerce recommendation method based on collaborative filtering technology." International Journal of Current Engineering and Technology 7.3 (2017): 974-982.

[16] Narayan, Vipul, et al. "To Implement a Web Page using Thread in Java." (2017).

[17] Srivastava, Swapnita, and P. K. Singh. "HCIP: Hybrid Short Long History Table-based Cache Instruction Prefetcher." International Journal of Next-Generation Computing 13.3 (2022).

[18] Srivastava, Swapnita, and P. K. Singh. "Proof of Optimality based on Greedy Algorithm for Offline Cache Replacement Algorithm." International Journal of Next-Generation Computing 13.3 (2022).

[19] Smiti, Puja, Swapnita Srivastava, and Nitin Rakesh. "Video and audio streaming issues in multimedia application." 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2018.

[20] Srivastava, Swapnita, and Shilpi Sharma. "Analysis of cyber related issues by implementing data mining Algorithm." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.

[21] Narayan, Vipul, and A. K. Daniel. "Multi-tier cluster based smart farming using wireless sensor network." 2020 5th international conference on computing, communication and security (ICCCS). IEEE, 2020