A

Week - 03

PROJECT REPORT ON

Text Summarization

Submitted by,

Snehal Gawade.

Niranjan Patil.

Mehul Patil.

Omkar Motale.

Hrithik Auchar.

Guidedby,

Mr. Chaitanya Patil

Prof. Rudragouda Patil.

School Of Computer Engg. & Technology

MIT Academy of Engineering

Dehu Phata, Alandi (D)

Pune - 412105, Maharashtra (India)

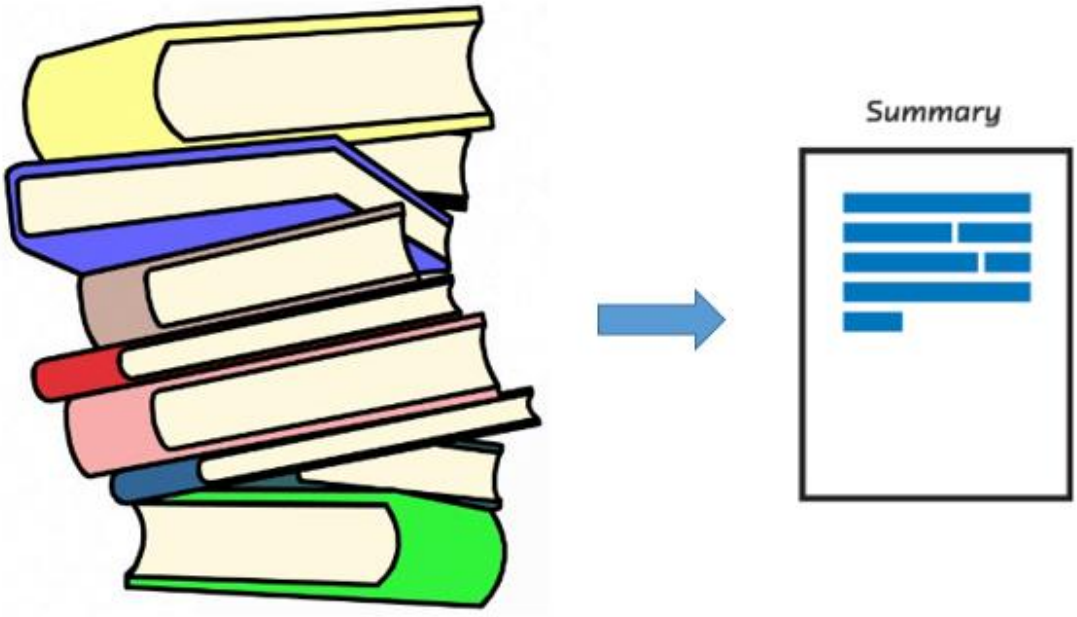
2019-2020

MIT | Academy of Engineering
Alandi (D), Pune – 412105

School Of Computer Engg. & Technology
(Accredited by NBA, ISO 9001:2008 Certified)

1. **Project Title:** Text Summarization.
2. **Purpose of Project:**

We all interact with an application which uses text summarization. Many of those applications are for the platform which publishes articles on daily news, entertainment, sports. With our busy schedule, we prefer to read the summary of that article before we decide to jump in for reading entire article. Reading a summary help us to identify the interest area, gives a brief context of the story. Summarization can be defined as a task of producing a concise and fluent summary while preserving key information and overall meaning.



3. **Objective of Project:** Summarizing the contents of a text file.
4. **Functional Requirements :**

1. Accept the file as the input for summarizing.
2. Pass it through the code and obtain a summary of the contents of text file.

5. Methodology:

How text summarization works?

In general there are two types of summarization, **abstractive** and **extractive** summarization.

1. **Abstractive Summarization:**

   Abstractive methods select words based on semantic understanding; even those words did not appear in the source documents. It aims at producing important material in a new way. They interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text.

It can be correlated to the way human reads a text article or blog post and then summarizes in their own word.

Input document → understand context → semantics → create own summary.

2. Extractive Summarization:

Extractive methods attempt to summarize articles by selecting a subset of words that retain the most important points.

This approach weights the important part of sentences and uses the same to form the summary. Different algorithm and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.

Input document → sentences similarity → weight sentences → select sentences with higher rank.

The limited study is available for abstractive summarization as it requires a deeper understanding of the text as compared to the extractive approach.

Purely extractive summaries often times give better results compared to automatic abstractive summaries. This is because of the fact that abstractive summarization methods cope with problems such as semantic representation, inference and natural language generation which is relatively harder than data-driven approaches such as sentence extraction.

EXTRACTIVE SUMMARIZATION METHODS:

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

B. Cluster based method:

C. Graph theoretic approach:

D. Machine Learning approach:

E. Text summarization with neural networks :

F. Automatic text summarization based on fuzzy logic :

1 Term Frequency-Inverse Document Frequency (TF-IDF) method:

It is a numerical statistic which reflects how important a word is in a given document. The TF-IDF value increases proportionally to the number of times a word appears in the document. This method mainly works in the weighted term-frequency and inverse sentence frequency paradigm .where sentence-frequency is the number of sentences in the document that contain that term. These sentence vectors are then scored by similarity to the query and the highest scoring sentences are picked to be part of the summary. Summarization is query-specific. The hypothesis assumed by this approach is that if there are ‘‘more specific words’’ in a given sentence, then the sentence is relatively more important. The target words are usually nouns .This method performs a comparison between the term frequencies (tf) in a document -in this case each sentence is treated as a document and the document frequency (df), which means the number of times that the word occurs along all documents. The TF/IDF score is calculated as follows:

2 Cluster based method:

In this method, the semantic nature of a given document is captured and expressed in natural language by a set of triplets (subjects, verbs, objects related to each sentence).Cluster these triplets using similar information. The triplet’s statements are considered as the basic unit in the process of summarization. More similar the triplets are, the more the information is useless repeated; thus, a summary may be constructed using a sequence of sentences related the computed clusters.

Example:

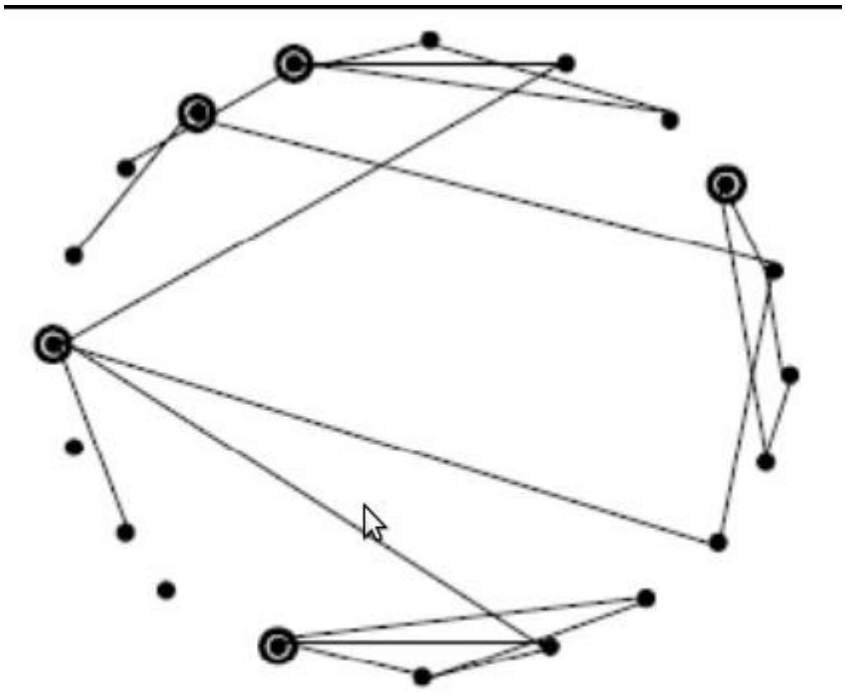
Cluster 1	⟨ambulance, be, scene⟩ ⟨police, be, scene⟩ ⟨ambulance, stream, area⟩ ⟨people, be, street⟩ ⟨pedestrian, peer, sky⟩
Cluster 2	⟨minister, inform, crash⟩ ⟨president, inform, incident⟩ ⟨spokesman, say, minister⟩
Cluster 3	⟨airplane,crash,tower⟩ ⟨plane, strike, building⟩ ⟨clock, fall, floor⟩ ⟨terror, attack, Washington⟩ ⟨terror, attack, New York⟩

Clusters of Triplets

3 Graph theoretic approach:

In this technique, there is a node for every sentence . Two sentences are connected with an edge if the two sentences share some common words, in other words, their similarity is above some threshold. This representation gives two results :The partitions contained in the graph (that is those sub-graphs that are unconnected to the other sub graphs), form distinct topics covered in the documents. The second result by the graph-theoretic method is the identification of the important sentences in the document. The nodes with high cardinality (number of edges connected to that node), are the important sentences in the partition, and hence carry higher preference to be included in the summary.

Figure shows an example graph for a document. It can be seen that there are about 3-4 topics in the document; the nodes that are encircled can be seen to be informative sentences in the document, since they share information with many other sentences in the document. The graph theoretic method may also be adapted easily for visualization of inter and intra document similarity.



Graph Theoretic Approach

4 Machine Learning approach:

In this method, the training dataset is used for reference and the summarization process is modeled as a classification problem: sentences are classified as summary sentences and non-summary sentences based on the features that they possess. The classification probabilities are learnt statistically from the training data, using Bayes' rule:

$$P(s \in S \mid F1, F2, ..., FN) = P(F1, F2, ..., FN \mid s \in S) * P(s \in S) / P(F1, F2, ..., FN)$$

where, **s** is a sentence from the document collection, F1, F2...FN are features used in classification. **S** is the summary to be generated, and P (s∈< S | F1, F2, ..., FN) is the probability that sentence s will be chosen to form the summary given that it possesses features F1,F2...FN.

5. Text summarization with neural networks:

In this method, each document is converted into a list of sentences. Each sentence is represented as a vector [f1,f2,...,f7], composed of 7 features.

Seven Features of a Document

1) f1 Paragraph follows title

2) f2 Paragraph location in document

3) f3 Sentence location in paragraph

4) f4 First sentence in paragraph

5) f5 Sentence length

6) f6 Number of thematic words in the sentence

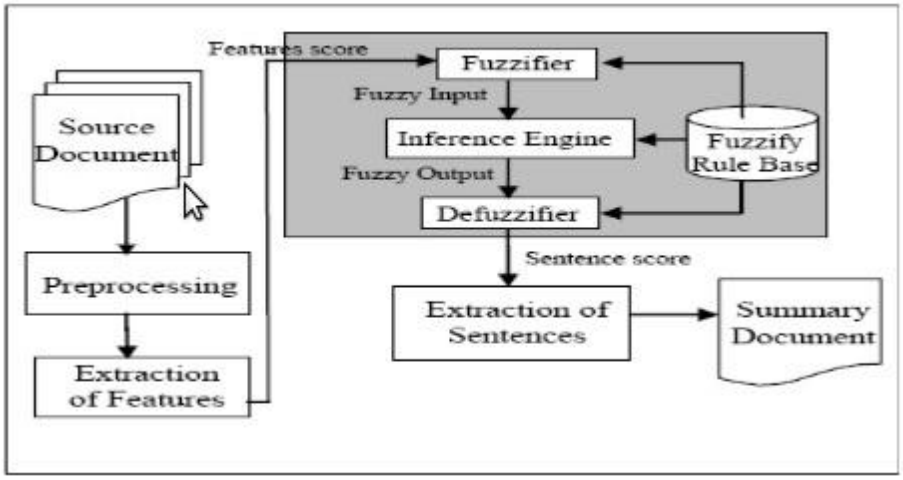
7) f7 Number of title words in the sentence

The first phase of the process involves training the neural networks to learn the types of sentences that should be included in the summary. Once the network has learned the features that must exist in summary sentences, we need to discover the trends and relationships among the features that are inherent in the majority of sentences. This is accomplished by the feature fusion phase, which consists of two steps: 1) eliminating uncommon features; and 2) collapsing the effects of common features.

6. Automatic text summarization based on fuzzy logic:

This method considers each characteristic of a text such as sentence length, similarity to little, similarity to key word and etc. as the input of fuzzy system .Then, it enters all the rules needed for summarization, in the knowledge base of system. Afterward, a value from zero to one is obtained for each sentence in the output based on sentence characteristics and the available rules in the knowledge base. The obtained value in the output determines the degree of the importance of the sentence in the final summary. The input membership function for each feature is divided into three membership functions which are composed of insignificant values (low L), very low

(VL), medium (M), significant values (High h) and very high (VH). The important sentences are extracted using IF-THEN rules according to the feature criteria.



The fuzzy logic system consists of four components: fuzzifier, inference engine, defuzzifier, and the fuzzy knowledge base. In the fuzzifier, crisp inputs are translated into linguistic values using a membership function to be used to the input linguistic variables. After fuzzification, the inference engine refers to the rule base containing fuzzy IFTHEN rules to derive the linguistic values. In the last step, the output linguistic variables from the inference are converted to the final crisp values by the defuzzifier using membership function for representing the final sentence score.

There are many techniques available to generate extractive summarization. To keep it simple, I will be using an unsupervised learning approach to find the sentences similarity and rank them. One benefit of this will be, you don't need to train and build a model prior start using it for your project.

It's good to understand **Cosine similarity** to make the best use of code you are going to see. **Cosine similarity** is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Since we will be representing our sentences as the bunch of vectors, we can use it to find the similarity among sentences. Its measures cosine of the angle between vectors. Angle will be **0** if sentences are similar.

It is important to understand that we have used **text rank** as an approach to rank the sentences. Text Rank is a general purpose graph-based ranking algorithm for NLP.

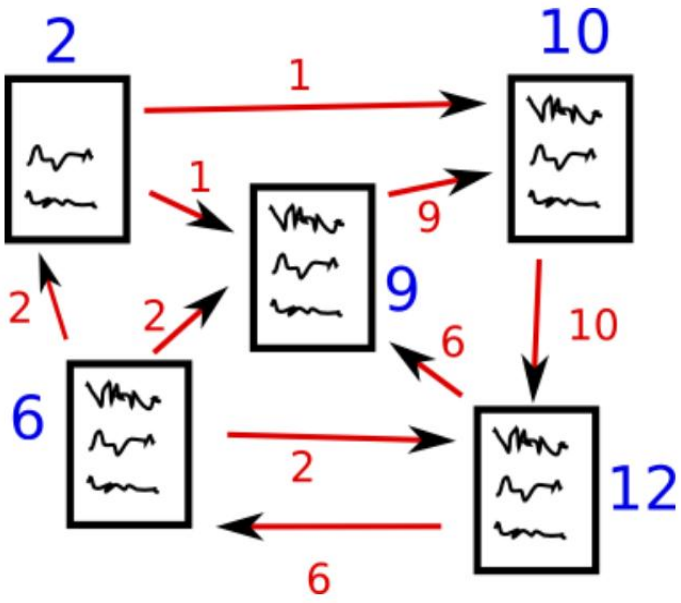
Code flow:

Input article → split into sentences → remove stop words → build a similarity matrix → generate rank based on matrix → pick top N sentences for summary.

Text Rank Algorithm:

Before getting started with the Text Rank algorithm, there's another algorithm which we should become familiar with – the PageRank algorithm. In fact, this actually inspired Text Rank! **PageRank is used primarily for ranking web pages in online search results.** Let's quickly understand the basics of this algorithm with the help of an example.

PageRank Algorithm



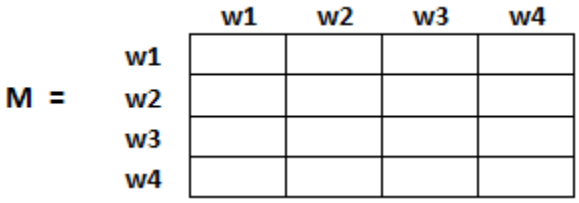
Suppose we have 4 web pages — w1, w2, w3, and w4. These pages contain links pointing to one another. Some pages might have no link – these are called dangling pages.

webpage	links
w1	[w4, w2]
w2	[w3, w1]
w3	[ ]
w4	[w1]

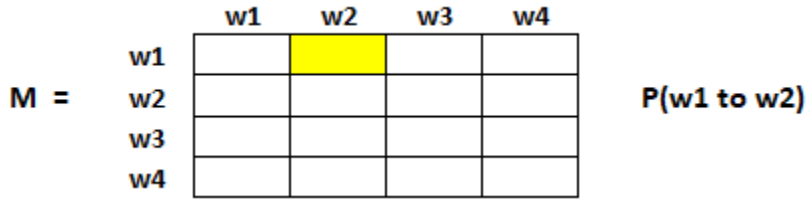
- Web page w1 has links directing to w2 and w4
- w2 has links for w3 and w1
- w4 has links only for the web page w1
- w3 has no links and hence it will be called a dangling page

In order to rank these pages, we would have to compute a score called the **PageRank score**. This score is the probability of a user visiting that page.

To capture the probabilities of users navigating from one page to another, we will create a square **matrix M**, having n rows and n columns, where **n** is the number of web pages.



Each element of this matrix denotes the probability of a user transitioning from one web page to another. For example, the highlighted cell below contains the probability of transition from w1 to w2.



The initialization of the probabilities is explained in the steps below:

1. Probability of going from page i to j, i.e., M[i][j], is initialized with **1/(number of unique links in web page wi)**
2. If there is no link between the page i and j, then the probability will be initialized with **0**
3. If a user has landed on a dangling page, then it is assumed that he is equally likely to transition to any page. Hence, M[i][j] will be initialized with **1/(number of web pages)**

Hence, in our case, the matrix M will be initialized as follows:

M =

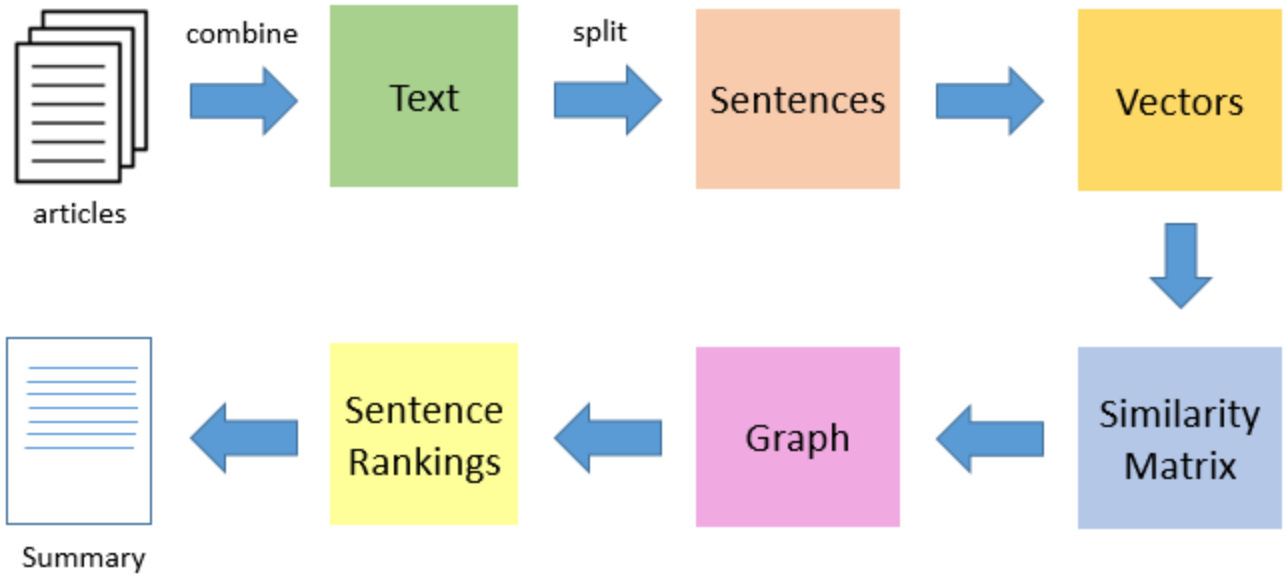
	w1	w2	w3	w4
w1	0	0.5	0	0.5
w2	0.5	0	0.5	0
w3	0.25	0.25	0.25	0.25
w4	1	0	0	0

Finally, the values in this matrix will be updated in an iterative fashion to arrive at the web page rankings.

Let's understand the TextRank algorithm, now that we have a grasp on PageRank. I have listed the similarities between these two algorithms below:

- In place of web pages, we use sentences
- Similarity between any two sentences is used as an equivalent to the web page transition probability
- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank

TextRank is an extractive and unsupervised text summarization technique. Let's take a look at the flow of the TextRank algorithm that we will be following:



- The first step would be to concatenate all the text contained in the articles
- Then split the text into individual sentences
- In the next step, we will find vector representation (word embeddings) for each and every sentence
- Similarities between sentence vectors are then calculated and stored in a matrix
- The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation
- Finally, a certain number of top-ranked sentences form the final summary.

6. Future Scope:

1. Multiple domain text summarization
2. Single document summarization
3. Cross-language text summarization (source in some language and summary in another language)