

1) There are two main strategies explained in this article . One is Abstractive summarisation and Extractive summarisation.

Extractive summarisation: The main aim of this method is to make the text document as summarised in an efficient manner . It extracts the most relevant and informative content directly from the original text document (input). It doesn't generate new sentences . It uses original sentences in the input document.

Abstractive summarisation:

It summarises the document based on the semantic understanding of the source document. It also extracts most relevant information even if those words didn't appear in the document. It interprets the text document using natural language processing techniques to generate summarization of the text document. It mimics the way humans read an article and then summarises in their own words.

2) The feature used in the text summarisation code is the similarity between the sentences. The code calculates the similarity using the cosine distance method.

- 1) First function tells about reading the text document and cleaning the sentences
- 2) Created a similarity matrix with empty zeros. Matrix shape is equal to the number of sentences.
- 3) Sentence similarity calculation for each pair of sentences (idx1 and idx2) the code calculates the similarity using sentence similarity function.
- 4) Then it computes the cosine distance between the sentences based on their word vectors.
- 5) It calculates cosine angle between the two vectors ranging from 0 to 1. It assigns the similarity score for the similarity matrix.

3) Similarity measurement used in this code is cosine similarity. It is a common metric method used to find the similarity between the two vectors. In the context of text summarisation this metric utilised to measure similarity between the pair of sentences.

The cosine similarity is calculated based on the cosine of the angle between two vectors, representing the sentences in a vector space. The formula for cosine similarity is:

$$\text{cosine\_similarity} = (A \cdot B) / (\|A\| * \|B\|)$$

4) We can use the Tf-idf method in this code. Tf-idf is the feature extraction method in natural language processing . It is mainly used for feature representation for the text. It can certainly be incorporated into the code to enhance the sentence similarity calculation and improve the quality of summary.

Instead of directly using the word frequencies we can use tf-idf then compute the similarity between the sentences. We can utilise cosine similarity with the TF-IDF vectors of the sentences to measure their similarity.

Tf-idf method provided by scikit-learn to compute the tf-idf values for each word in the sentences. It assigns weight to each word based on frequency in the sentence and its importance in the entire source document.

5)Outputs are not the same , Each text document contains different content . When comparing the individual inputs and outputs of the text document . Summarisation of the outputs are different.Based on the outputs summarised text captures the main highlights of the original article . The first document talking about the cloud hub programs by microsoft aims to empower the students with new AI skills. It is focusing on the program's objectives,support provided to institutions, the significance of AI skills, and related initiatives by Microsoft.

Second outputs of the summarised text talking about the Trump administration's decision to withdraw troops from Afghanistan and its implications for the ongoing conflict. It is clearly concise , well structured and efficient . I observed that lengths of document input and outputs are different .

All the outputs of text document demonstrates the ability to distil important information into a condensed form. It encompasses the main concepts, crucial details, and key arguments while presenting them in a clear and coherent manner. These well-summarised documents offer a comprehensive snapshot of the original text, allowing readers to grasp the main content and understand its significance without needing to go through the entire document.