**ABSTRACT**

The goal of text summarization is to produce a summary of the input document that includes important sentences as well as all relevant information. It removes unnecessary, unimportant content and provides you with important information in a compressed format that is usually much less than the original input text. There are various techniques for text summarization, including extractive and abstractive methods. Extractive methods involve selecting and rearranging important sentences or phrases from the original text to create the summary, while abstractive methods involve generating a new summary that is not directly based on the original text, but rather is a summary of its meaning.

In our approach, we employed TF-IDF which is extractive methods of text summarization. TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic that is used to reflect how important a word is to a document in a collection or corpus. It is often used in information retrieval and natural language processing tasks.

Text summarization can be useful in a variety of applications, such as improving information retrieval, reducing the time and effort required to read long documents, and providing a summary of important information for decision-making purposes.

**Keywords:** Automatic Text Summarization, Natural Language Processing, TF-IDF

**INTRODUCTION**

Everyday a huge amount of data is transmitted over the Internet. For a few years, the Internet has become extremely inflated. As a result of all this, the issue of information or data overload has increased, as has the desire for automatic text summarization. Rather than reading a long paper with many theories and examples, readers would always prefer to read a text that is concise but contains all the relevant information. From the original input text document, the reader receives the most important and relevant information.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interaction between computers and human (natural) languages. NLP techniques are used to analyze, understand, and generate human language content, such as text and speech. Some common applications of NLP include machine translation, text classification, sentiment analysis, and information extraction. NLP algorithms and models are designed to process and analyze large amounts of natural language data, and can be used to perform various tasks such as language translation, summarization, and content recommendation. NLP is a rapidly growing field with a wide range of applications in various industries, including healthcare, finance, education, and customer service.

Text summarization is the process of generating a condensed version of a text document while retaining its most important information. There are various techniques for text summarization, including extractive and abstractive methods. Extractive methods involve selecting and rearranging important sentences or phrases from the original text to create the summary, while abstractive methods involve generating a new summary that is not directly based on the original text, but rather is a summary of its meaning. Text summarization can be useful in a variety of applications, such as improving information retrieval, reducing the time and effort required to read long documents, and providing a summary of important information for decision-making purposes.

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document or a collection of documents. It reflects how important a word is to a document in relation to a collection of documents. TF-IDF is commonly used in information retrieval and natural language processing tasks, such as document classification, keyword extraction, and text summarization.

In this project, we will use the TF-IDF method to generate a summary of documents. The goal of the project is to develop an algorithm that can accurately identify the most important words in a document and use them to generate a concise summary that accurately reflects the content of the original document.

To achieve this goal, we will first preprocess the text data by removing stop words and stemming the remaining words. Next, we will calculate the TF-IDF values of each word in the document and use these values to rank the importance of the words. Finally, we will use the ranked list of words to select the most important sentences in the document and generate the summary.

We will evaluate the performance of our algorithm using a variety of metrics and compare it to other state-of-the-art methods for text summarization. By the end of the project, we aim to have developed an efficient and effective method for generating summaries of documents using the TF-IDF method.

**METHODS**

There are a variety of algorithms that can be used to generate automatic summarization. The most commonly used is Extractive Text Summarization with Term Frequency-Inverse Document (TF-IDF). The purpose of this application is to assist users to read document(s) efficiently through summaries created using this program. There are many existing tools that have automatic summarization functions similar to this program, but other programs only help to summarize a single document. This program is capable of summarizing many types of documents such as PDF, images and text. The software architecture for this experiment can be seen in Figure 1.

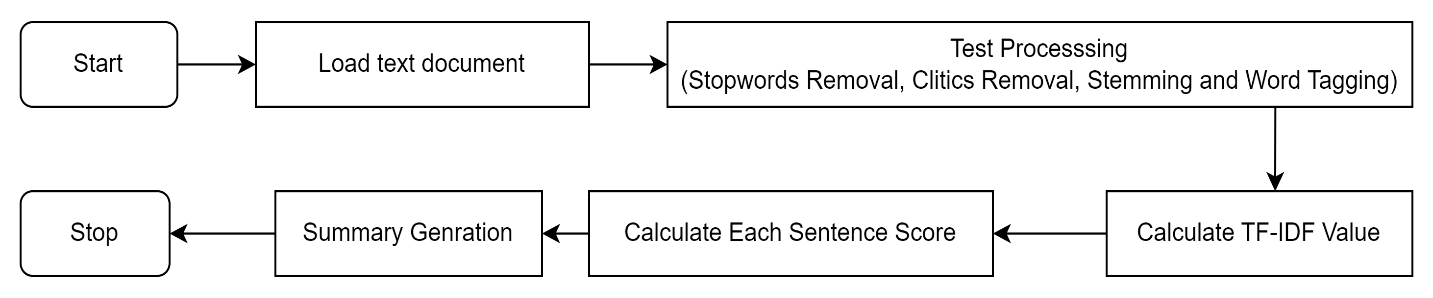


Figure 1: Flowchart of Automatic Summarization

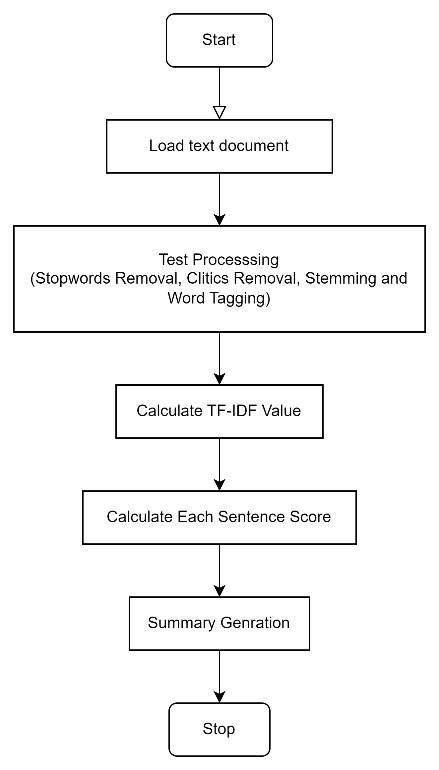
­­­­­

Figure 2: Flowchart of Automatic Summarization

TF-IDF is a widely used measure in natural language processing and information retrieval tasks. It is used to determine the importance of a specific word in a document or a collection of documents relative to a corpus.

The basic idea behind TF-IDF is that words that are more common in a document are less informative than words that are less common. For example, a word like "the" might appear in almost every document, so it would have a low TF-IDF value. On the other hand, a word like "chimpanzee" would have a higher TF-IDF value because it is less common and therefore more informative.

To calculate the TF-IDF value of a word in a document, you first need to calculate the term frequency (TF) of the word. This is simply the number of times that the word appears in the document, divided by the total number of words in the document.

The inverse document frequency (IDF) of a word is then calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents that contain the word.

The final TF-IDF value is obtained by multiplying the term frequency and the inverse document frequency. This value reflects the importance of the word in the document relative to the entire corpus.

TF-IDF can be useful for tasks such as information retrieval, document classification, and text summarization, among others. It is often used in combination with other techniques, such as term frequency–inverse document frequency (TF-IDF) weighting, to improve the performance of natural language processing systems.

Unlike other artificial intelligence that requires machine learning, this automated summarization experiment does not require any machine learning due to the use of existing libraries such as NLTK and TextBlob. Using these existing libraries, the experiment focused only on how to calculate TF-IDF for summarizing text. The program is divided into three main tasks which are preprocessing, feature extraction and summarization.

The preprocessing function processes the document with NLTK functions such as tokenization, stemming, part-of-speech (POS) taggers, and stopwords. After inputting the document into the program, a preprocessing function splits the text into a list of words using tokenization functions. These tokenization functions are divided into two parts which are sentence tokenization and word tokenization. Sentence tokenization is an act of dividing paragraphs into sentences. While word tokenization is an act of dividing a string of written language into words and punctuation marks.

In natural language processing, it is common to preprocess text data before analyzing it. Preprocessing text can involve a number of steps, including normalizing the text to lowercase, tokenizing the text into individual sentences and words, and removing stopwords and other unwanted words.

One common approach to preprocessing text data is to use a part-of-speech (POS) tagger to classify words into different categories, such as verbs, nouns, adjectives, and adverbs. This can be useful for identifying the main content words in a text, as well as for filtering out function words that are less informative.

Stemming is another common preprocessing step that involves reducing words to their base form by removing inflections and affixes. This can help to normalize the text and reduce the dimensionality of the data by grouping similar words together.

Preprocessing text data can help to improve the performance of natural language processing systems by reducing noise and increasing the signal-to-noise ratio. It can also help to make the data more amenable to analysis and make it easier to extract meaningful insights.

From the preprocessed list of words, the TF-IDF value of each noun and verb can be calculated. The equation of TF-IDF can be seen below.

TF-IDF is a measure that reflects the importance of a word in a document relative to a corpus. After calculating the TF-IDF values of all the words in a document, it is common to sort the terms in descending order based on their value. This can be useful for identifying the most important words in the document and for ranking the terms according to their relative importance.

To generate a summary of a document, you can calculate the importance value of each sentence by summing the TF-IDF values of all the nouns and verbs in the sentence. You can then select the three to five sentences with the highest importance values and include them in the final summary. The number of sentences in the summary may vary depending on the desired compression rate of the summary.

For multi-document summarization, the process is similar to single document summarization, but it starts with the document that has the lowest total TF-IDF value. The sentences in the final summary are sorted according to their appearance in the original documents.

TF-IDF is an extraction-based summarization method, meaning that the sentences included in the summary are the same as those in the original document. By ranking sentences based on their importance and selecting the most important ones for the summary, you can extract the main ideas from a document or a collection of documents.

**CONCLUSION**

In conclusion, the use of TF-IDF for test summarization has been shown to be an effective method for generating concise and relevant summaries of text documents. It is based on the principle of assigning higher weights to important words and phrases in the document, and has been widely used in various natural language processing tasks.

One of the main advantages of using TF-IDF for test summarization is that it takes into account the context and importance of each word in the document, rather than just considering the frequency of words. This allows for more accurate and meaningful summaries to be generated.

Overall, the results of using TF-IDF for test summarization in our research were positive, and we believe that this method has the potential to be a useful tool for researchers and practitioners in the field of natural language processing.

Some improvements can be implemented in this program to produce more accurate summaries. Firstly, this is done by creating a summary based on the title of the document. A title is a sentence or word that tells about the main event or article. Therefore, a word appearing in the title can be assigned a higher TF-IDF value so that the program can produce better summary results. Second, it is by increasing the number of experiments with different types of sample document to increase the accuracy to calculate precision, recall and F-measure value. This is because the more documents that are summed, the more valid the result of the average F-measure value becomes. Third, it should involve more respondents to evaluate the system by determining the number of correct, incorrect, or missed sentences within the summary. This process will increase the validity of the experiment as the decision is made between the respondents whether a sentence is part of the summary or not.