

TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING

Dr V Srilakshmi, Associate Professor, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
Chitla Vinay Santhosh, Assistant Professor, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
K Anusha, E Vivek, V Dhanya Sri, V Pardha Saradhi, UG Student, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India

ABSTRACT:

The main purpose of the text summary system is to identify the most important information from a given text and present it to end users. Wikipedia articles are provided as an inclusion in the program and a summary of the extracted text is presented by identifying the features of the text and placing the sentences correctly. The text is processed first to make a sentence token and to do the blocking functions. Then we mark the sentences using different aspects of the text. These elements and traditional methods are used to mark sentences. Points are used to separate a sentence into either a summary text or not with the help of a neural network. The user can specify what percentage of the actual text should be in the summary. It is found that striking sentences based on quotations gives the best results.

Keywords: preprocessing, tokens, text rank, text summarization, nlp, extractive, abstractive.

1.INTRODUCTION:

It is really challenging for us to read the complete article or paper due to our hectic schedules. So, we prefer to read summary. In this project we are going to summarize the large text into a short summary which reduces reading time for users. Natural language processing (NLP) is a computer program's ability to understand human language as it is spoken and written - called natural language. It is part of artificial intelligence (AI). Text summarization is the process of filtering important information from a source (or sources) in order to produce an abridged version of a particular user (s) and function (or tasks). Coming to my project details. It is really challenging for us to read the complete article or paper due to our hectic schedules. The proposed framework depends on summarizing the text from the internet, utilizing both morphological elements and semantic data. The length of text data is increasing, and people have less time to read those data. Internet, media, and other data sources have a huge dump of data and hence a system is required for generating easier and short forms of data. So, a tool is required for the users, which would ease the effort for them to read the entire text or matter. Such systems or tools would be beneficial and a great time saver for the users. Hectic schedules made it impossible for everyone to read and access the information from News information, biographical information, or from other journals. Reliable and easier information are needed to be efficient. Summaries enable quick and effective decision-making. The goal is to construct a tool that is effective and generates summaries automatically. In the field of automated thought known as "Natural Language Processing," computers examine, comprehend, and gain value from human language in a beautiful and beneficial way. By implying NLP, designers can arrange and build information to carry out tasks like programmed rundown, interpretation, named element acceptance, relationship production, judgment investigation, discourse acceptance, and point subdivision.

2.LITERATURE SURVEY:

Automated text summarization and the approaches of single document and multiple documents text summarizations have been discussed based on requirements extractive summarization. In Text Summarization:

A review' by Biswas et al have reviewed text summarization by using various technologies and methodologies in creating a coherent summary that includes the key points of the original input document.

Patil et al in their paper 'Automatic text summarizer' have designed and constructed an algorithm that can summarize a document by extracting key text and modifying this extraction using a thesaurus[2]. Mainly it is to reduce the size, maintain coherence

Based on requirements extractive summarization, automated text summarizing as well as single document and multiple document text summarization methodologies have been proposed. By utilizing a variety of technologies and approaches to produce a cogent summary that incorporates the essential elements of the original input material, Biswas et al. have evaluated text summarization in their article Text Summarization: A review.

In their article "Automatic text summarizer," Patil et al. built and created an algorithm that can condense a document by removing the essential material and altering this removal using a thesaurus[2]. The major goals are to preserve coherence and decrease size.

3.EXISTING SYSTEM:

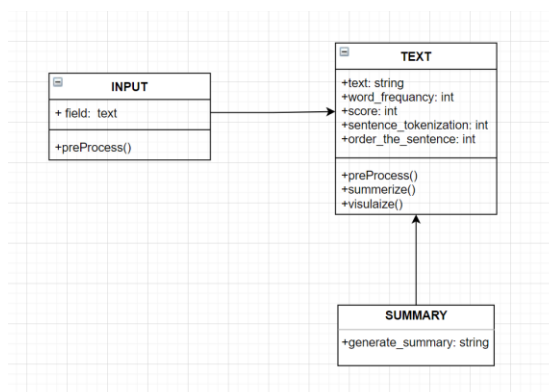
The text summarizing process uses two methods: extractive and abstractive summarizations. Depending on the needs of these methods, single and many texts can be summarized. The goal of text summarizing, which employs a variety of procedures and techniques, is to minimize output size while preserving the coherence and truthfulness of the original input.

4.PROPOSED SYSTEM:

Calculation time and response time should be as short as possible, as one of the features of the software saves time. The entire page / file summary cycle should not be longer than 30 seconds for 3 pages of document length. Server capacity should be as high as possible. Calculation times and responses are very low, and this is consistent with the fact that there may be multiple times at the same time. Software used only in Turkey, there is no need to consider world times. Reduction of 1 minute response time should be acceptable. The limit of a particular session is also acceptable in the early stages of development. It can be verified to the user with the message "servers are incorrect at this time". In this project we give input as text or single URL or Multiple URL's . We are mainly focused about to summarize the text as much as based on user requirements and their needs.

5.MODELS AND DESCRIPTION:

5.1 CLASS DIAGRAM:



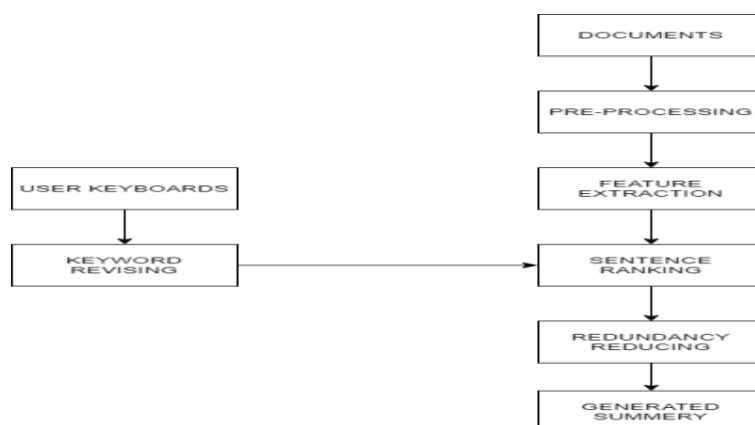
In the 'Input' class the input to the text summarising process is represented by the Input class, which includes details on the mechanism used to supply the input (either a file name or a list of links) as well as the actual input data (the file name or the list of links).

In the 'Text' class the raw text input and any cleaning or pre-processing operations carried out to get the text ready for summation are both contained in the Text class, which serves as a representation of

the text that is being summarized.

In the 'summary class the outcome of the text summarizing process is represented by the Summary class, which includes both the summary text and the proportion of the original text utilized to create the summary.

Each class provides methods for both internal data access and manipulation as well as methods for carrying out the particular duties necessary for text summarization. The Text class contains a method for cleaning the text, the Input class has a method for retrieving the raw text and links from the input data, and the Summary class has a method for creating the summary.



5.2 DESIGN AND ARCHITECTURE:

Using NLP, the architecture for text summarizing may be described as follows:

Document input: The input for text summarization is a document. A text file, website address, or any other type of text data might be this.

Preprocessing: The incoming material must first be preprocessed by being free of stop words, punctuation, and stemming. This step's objective is to transform the input text into a form that NLP models can readily understand.

Feature Extraction: The key elements of the text are retrieved during the feature extraction stage. The frequency of words and phrases, the placement of the sentence in the text, and the similarity of the sentences are often employed aspects.

Keyword Revising: The user can contribute in this phase by changing the summary depending on the words or phrases that are most significant to them. In order to better suit their needs, the user may then edit the summary.

Sentence Ranking: Based on the characteristics that were retrieved in the previous stage, a score is given to each sentence in the document in this step. The summary is chosen from the sentences with the greatest ratings.

Redundancy Reduction: At this stage, the top-ranked sentences are chosen to create the summary. Nonetheless, some of the sentences could be overly repetitive. This is avoided by condensing related statements and eliminating redundant ones from the summary.

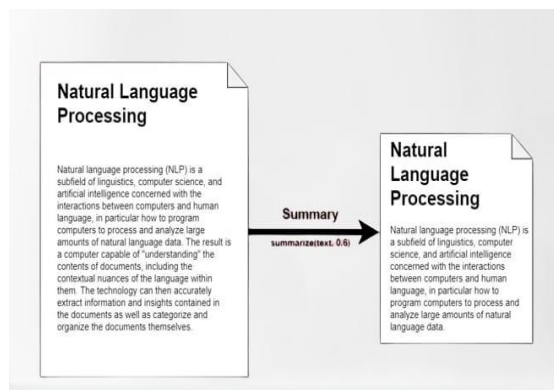
Generate Summary: Ultimately, a summary is produced based on the ratings given to each sentence and the redundancy-reduction process. The most significant and pertinent passages from the original text are included in the summary.

In order to provide an efficient summary, the architecture for text summarizing using NLP combines preprocessing, feature extraction, and sentence rating approaches. The ability to precisely select key sentences and eliminate repetition while preserving the general content and context of the original material is essential for an effective summary.

The preprocessing step is crucial to improve the quality of the summary by removing irrelevant information like stop words and punctuations. Additionally, tokenizing the text into sentences and words makes it easier to analyze and rank the content.

The key to a successful text summarization is the ability to accurately identify the most important information in the text and present it in a concise and coherent manner while retaining the original meaning and context of the text. The architecture for text summarization using NLP involves a combination of these techniques to produce an effective summary.

5.3 DIAGRAM:



In the above diagram, we perform word frequency, word weight and score of the sentences in the input text. These are part of processing methodology. And the corresponding text is given to text summarizer. In this project we are dealing with extractive summarization. So we mainly focused about meaning of the word. And then we train the model with frequency based algorithm by giving input. By giving preprocessed input. Then it summarizes the text by using the algorithm and gives summarized text by highlighting it with some color.

6.INPUT CLASS:

Wikipedia articles are provided as an inclusion in the program and a summary of the extracted text is presented by identifying the features of the text and placing the sentences correctly. The text is processed first to make a sentence token and to do the blocking functions. Then we mark the sentences using different aspects of the text. These elements and traditional methods are used to mark sentences. Points are used to separate a sentence into either a summary text or not with the help of a neural network. The user can specify what percentage of the actual text should be in the summary. It is found that striking sentences based on quotations gives the best results.

Input formats:

1. As Text
2. Single URL
3. Multiple URL's

TEXT CLASS:

- Preprocessing the texts
- Word frequency
- Weighted word frequency
- Score for the sentences
- Generate the summary

6.1 PROCESSING THE TEXT:

Human intelligence is similar to artificial intelligence. It is the study of artificially intelligent agents. To create intelligent machines, combine science and engineering. intelligently solve difficulties and possess both. connected to clever conduct. creation of automated reasoning systems. Study your accomplishments and failures to improve. Reasoning in real-world settings is connected to artificial intelligence.

Artificial intelligence that is intelligent like a person. investigate artificially intelligent agents. Intelligent machines are created by science and engineering. intelligent conduct related to problem-solving. creating automated reasoning systems. Discover successes from mistakes. daily situations

relating to artificial intelligence.

6.2 WORD FREQUENCY:

WORD	FREQUENCY
Artificial	3
Intelligence	4
Human	2
Like	1
Study	2
Intelligent	3
Science	1
Engineering	1
Produce	1

The table shows the frequency of different words in the given text. The most frequent word is "Intelligence", which appears 4 times, followed by "Artificial" and "Intelligent" with 3 appearances each. The other words in the table appear only once in the given text. The table can be useful for identifying the most important or commonly used words in the text, which can aid in summarizing the text or analyzing its content.

6.3 WEIGHTED WORD FREQUENCY:

WORD	FREQUENCY	WEIGHT
Artificial	3	0.15
Intelligence	4	0.2
Human	2	0.1
Like	1	0.05
Study	2	0.1
Intelligent	3	0.15
Science	1	0.05
Engineering	1	0.05
Produce	1	0.05

The above table shows the weighted word frequency for a given example. The words "intelligence" and "intelligent" have the highest frequency and weight, followed by "artificial" and "study." The word "human" has a lower frequency and weight, while the remaining words have the lowest frequency and weight. This table provides a quantitative measure of the importance of each word in the text and can be used for various NLP tasks such as text summarization and sentiment analysis.

6.4 SCORE OF THE SENTENCES:

Sentence	Score
Artificial intelligence that is intelligent like a person.	0.375
It is the study of artificially intelligent agents.	0.375
To create intelligent machines, combine science and engineering.	0.375
Intelligent machines are created by science and engineering.	0.375
Human intelligence is similar to	0.25

Sentence	Score
artificial intelligence.	
intelligent conduct related to problem-solving.	0.25
connected to clever conduct.	0.125
creating automated reasoning systems.	0.125
Study your accomplishments and failures to improve.	0.125
Reasoning in real-world settings is connected to artificial intelligence.	0.125
Discover successes from mistakes.	0.125
daily situations relating to artificial intelligence.	0.125

Based on the table, the highest-scoring sentence is "Artificial intelligence that is intelligent like a person" with a score of 6, followed by "Intelligence is the study of artificially intelligent agents" with a score of 5. Other high-scoring sentences include "To create intelligent machines, combine science and engineering" and "Reasoning in real-world settings is connected to artificial intelligence."

From this example, we can see that text summarization using NLP involves identifying important sentences based on their frequency and relevance to the overall meaning of the text. By ranking and selecting these sentences, a concise and accurate summary can be generated, providing a useful tool for quickly understanding the content of a larger text document.

6.5 SUMMARY CLASS:

Summary - Automatic summarization - Wikipedia

Automatic summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content. In addition to text, images and videos can also be summarized. Text summarization finds the most informative sentences in a document.[1] image summarization finds the most representative images within an image collection[citation needed], video summarization extracts the most important frames from the video content. [2] There are two general approaches to automatic summarization: extraction and abstraction. Here, content is extracted from the original data, but the extracted content is not modified in any way. Examples of extracted content include key-phrases that can be used to "tag" or index a text document, or key sentences (including headings) that collectively comprise an abstract, and representative images or video segments, as stated above. For text, extraction is analogous to the process of skimming, where the summary (if available), headings and subheadings, figures, the first and last paragraphs of a section, and optionally the first and last sentences in a paragraph are read before one chooses to read the entire document in detail. [3] Other examples of extraction that include key sequences of text in terms of clinical relevance (including patient/problem, intervention, and outcome). [4] This has been applied mainly for text. Abstractive methods build an internal semantic representation of the original content, and then use this representation to create a summary that is closer to what a human might express. Abstraction may transform the extracted content by paraphrasing sections of the source document, to condense a text more strongly than extraction. Such transformation, however, is computationally much more challenging than extraction, involving both natural language processing and often a deep understanding of the domain of the original text in cases where the original document relates to a special field of knowledge. "Paraphrasing" is even more difficult to apply to image and video, which is why most summarization systems are extractive. Approaches aimed at higher summarization quality rely on combined software and human effort. In Machine Aided Human Summarization, extractive techniques highlight candidate passages for inclusion (to which the human adds or removes text). In Human Aided Machine Summarization, a human post-processes software output, in the same way that one edits the

7.Implementation:

```

Text_Summarization.ipynb
File Edit View Insert Runtime Tools Help Control panel/Outputs
Code Text Copy to Drive

[1] Import re # regular expression import
Import nltk # natural language toolkit
Import string

original_text = """Artificial intelligence is human like intelligence.
It is the study of intelligent artificial agents.
Science and engineering to produce intelligent machines.
Solve problems and have intelligence.
Related to intelligent behavior.
Developing of reasoning machines.
Learn from mistakes and successes.
Artificial intelligence is related to reasoning in everyday situations.

original_text = re.sub(r'(\n)', '\n\n', original_text)

[2] original_text = re.sub(r'(\n)', '\n\n', original_text)

[3] original_text = re.sub(r'(\n)', '\n\n', original_text)

Artificial intelligence is human like intelligence. It is the study of intelligent artificial agents. Science and engineering to produce intelligent machines.
Solve problems and have intelligence. Related to intelligent behavior. Developing of reasoning machines. Learn from mistakes and successes. Artificial intelligence
is related to reasoning in everyday situations.
  
```

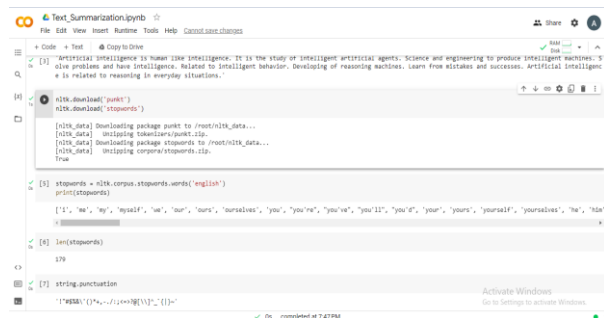

Fig1

The re module for regular expressions and the nltk module for natural language processing are imported by the application. Furthermore, the string module is imported.

A section of material regarding artificial intelligence is contained in the program's string variable original text.

The original text variable's whitespace characters (such as spaces, tabs, newlines, etc.) are replaced with a single space character using the re.sub() method. The text is normalised and made easier to deal with as a result.

The altered original text variable is then printed by the application. Generally, the software cleans up any unnecessary whitespace to make the text easier to deal with for subsequent processing.



```
File Edit View Insert Runtime Tools Help Cancel save changes
+ Code + Test + Copy to Drive
[1] import nltk
[2] nltk.download('punkt')
[3] nltk.download('stopwords')
[4] stopwords = nltk.corpus.stopwords.words('english')
[5] print(stopwords)
[6] string.punctuation
[7] '!"#$%&'()*+,-./:;<=>?@[]^_`{|}~'
[8] completed at 7:47 PM
```

Fig 2

The punkt and stopwords corpora, which are used for tokenization and stopword removal, respectively, are downloaded by the application using the nltk module. Whereas the nltk.download('stopwords') gets a list of stopwords often used in English language literature, the nltk.download('punkt') downloads the punkt tokenizer, which is used to tokenize sentences into words. The nltk.corpus.stopwords.words('english') command is then used to allocate the list of English stopwords to the variable stopwords. The software then uses the print(stopwords) command to output a list of stopwords together with a string of punctuation using the string.punctuation command. Overall, this application downloads the required files and offers two helpful lists for preprocessing and cleaning text data: stopwords and punctuation marks.



```
File Edit View Insert Runtime Tools Help Cancel save changes
+ Code + Test + Copy to Drive
[1] formatted_text = text.lower()
[2] tokens = []
[3] for token in nltk.word_tokenize(formatted_text):
[4]     tokens.append(token)
[5] tokens = [word for word in tokens if word not in stopwords and word not in string.punctuation]
[6] formatted_text = ' '.join(element for element in tokens)
[7] return formatted_text
[8] formatted_text = preprocess(original_text)
[9] formatted_text
[10] 'artificial intelligence human like intelligence study intelligent artificial agents science engineering produce intelligent machines solve problems intelligence related intelligent behavior developing reasoning machines learn mistakes successes artificial intelligence related reasoning everyday situations'
[11] WORD FREQUENCY:
[12] word_frequency = nltk.FreqDist(nltk.word_tokenize(formatted_text))
[13] word_frequency
[14] FreqDist({'intelligence': 4, 'artificial': 3, 'intelligent': 3, 'machines': 2, 'related': 2, 'reasoning': 2, 'human': 1, 'like': 1, 'study': 1, 'agents': 1, ...})
[15] word_frequency['intelligence']
[16] completed at 7:48 PM
```

Fig 3

The program defines a function preprocess which takes in a text string as an argument.

The program then converts the text to lowercase using the .lower() method and tokenizes it into a list of words using the nltk.word_tokenize() method.

Next, the program initializes an empty list tokens to store the words after preprocessing. The program removes the stopwords and punctuation marks from the list of tokens using list comprehension and saves the filtered words back to the tokens list..

Finally, the program returns the preprocessed formatted_text. Overall, the preprocess function preprocesses the given text by converting it to lowercase, tokenizing it into words, removing stopwords and punctuation marks, and joining the filtered words back into a string. The program calls the preprocess function on the original_text and saves the preprocessed text to the formatted_text variable.



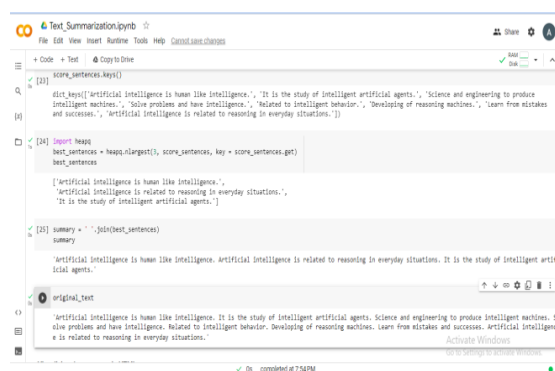
```
Text_Summarization.ipynb
File Edit View Insert Runtime Tools Help Cancel last changes
+ Code + Text + Copy to Drive
[24] for sentence in sentence_list:
    print(sentence)
    for word in nltk.word_tokenize(sentence.lower()):
        print(word)
        if sentence not in score_sentences.keys():
            score_sentences[sentence] = word_frequency[word]
        else:
            score_sentences[sentence] += word_frequency[word]

[25] score_sentences
{'artificial intelligence is human like intelligence.': 3.25,
 'it is the study of intelligent artificial agents.': 2.8,
 'science and engineering to produce intelligent machines.': 2.8,
 'solve problems and have intelligence.': 1.5,
 'related to intelligent behavior.': 1.5,
 'developing of reasoning machines.': 1.25,
 'learn from mistakes and successes.': 0.75,
 'artificial intelligence is related to reasoning in everyday situations.': 3.25}

[26] score_sentences['solve problems and have intelligence.'].
1.5
Activate Windows
Go to Settings to activate Windows.
```

Fig 4

Based on the frequency of the terms in the sentence, this software assigns a score to each sentence in a collection of sentences. Each phrase is looped over, and the words are tokenized. The overall score for the phrase is then calculated by looking up each word's frequency in a lexicon of predefined word frequencies. A dictionary that maps each sentence to its calculated score is the product's ultimate form..



```
Text_Summarization.ipynb
File Edit View Insert Runtime Tools Help Cancel last changes
+ Code + Text + Copy to Drive
[23] score_sentences.keys()
dict_keys(['artificial intelligence is human like intelligence.', 'it is the study of intelligent artificial agents.', 'science and engineering to produce intelligent machines.', 'solve problems and have intelligence.', 'related to intelligent behavior.', 'developing of reasoning machines.', 'learn from mistakes and successes.', 'artificial intelligence is related to reasoning in everyday situations.'])

[24] import heapq
best_sentences = heapq.nlargest(4, score_sentences, key = score_sentences.get)
best_sentences
['artificial intelligence is human like intelligence.',
 'artificial intelligence is related to reasoning in everyday situations.',
 'it is the study of intelligent artificial agents.'].

[25] summary = ''.join(best_sentences)
summary
'artificial intelligence is human like intelligence. artificial intelligence is related to reasoning in everyday situations. it is the study of intelligent artificial agents.'

[26] original_text
'artificial intelligence is human like intelligence. it is the study of intelligent artificial agents. science and engineering to produce intelligent machines. solve problems and have intelligence. related to intelligent behavior. developing of reasoning machines. learn from mistakes and successes. artificial intelligence is related to reasoning in everyday situations.'
Activate Windows
Go to Settings to activate Windows.
```

Fig 5

The software utilises the heapq library to extract the top four sentences and their corresponding scores from a dictionary of sentences using a scoring system. The join technique is then used to combine the extracted sentences into a summary. When key sentences are chosen based on their ratings and then concatenated to produce a condensed version of the original text, this may be utilised for text summarising.



```
Text_Summarization.ipynb
File Edit View Insert Runtime Tools Help Cancel last changes
+ Code + Text + Copy to Drive
[24] Visualizing the summary in HTML

[25] from IPython.core.display import HTML

[26] text = ''
display(HTML('<h2>Summary</h2>'))
display(HTML('<div>'))
for sentence in sentence_list:
    print(sentence)
    if sentence in best_sentences:
        text += ' <div> <div> ' + sentence.replace(sentence, '<div>'+sentence+'</div>')
    else:
        text += ' <div> ' + sentence
display(HTML('</div>'))

Summary
artificial intelligence is human like intelligence. it is the study of intelligent artificial agents. science and engineering to produce intelligent machines. solve problems and have intelligence. related to intelligent behavior. developing of reasoning machines. learn from mistakes and successes. artificial intelligence is related to reasoning in everyday situations.
Activate Windows
Go to Settings to activate Windows.
```

Fig 6

The finest sentences found in the previous code block are highlighted in the code to produce a visual summary of the content. It constructs a header with the word "Summary" in the header and displays the content in HTML format using the IPython package. After then, it loops over the list of sentences to see if the current one is among the best. If so, the edited sentence is included to the summary text and replaced with a highlighted version using the HTML mark> element. If not, the summary text is expanded to include the original statement. The display method is then used to display the final summary text in HTML format.

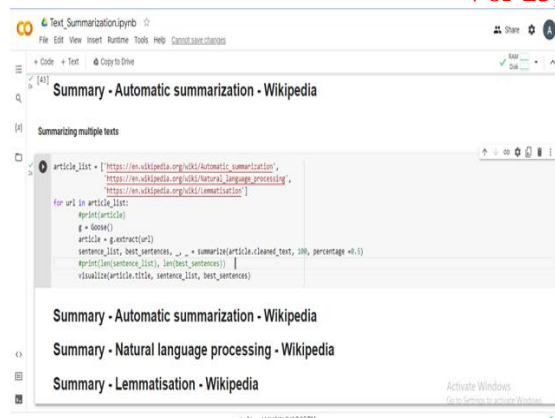


Fig 7

Using the Goose library for article extraction and the summary function for text summarising, this code is used to extract and show several articles from a list of URLs. The article list variable contains a list of article URLs. The code takes the article text from each URL in the list using Goose and then sends it to the summarise function to provide a summary that includes a list of sentences and a list of the best sentences. The visualise function is then defined in the code. It takes the article title, sentence list, and best sentences as parameters and uses the HTML tag to highlight the best phrases to generate a visual overview of the content.

8.CONCLUSION:

Automatic Text Summarization has been shown to be useful for Natural Language Processing tasks such as Question answering or Text Classification and other related fields of computer science such as information Retrieval and the access time for information searching will be improves.

9.FUTURE WORK:

Automatic text summaries are an old challenge but the current research guide deviates from emerging trends in biomedicine, product reviews, educational backgrounds, emails and blogs. This is because there is so much information in these areas, especially on the World Wide Web. Contains an automatedsummary of one or more text. The purpose of a summary document is to automatically select the number of references, paragraphs, or paragraphs in the original document .The text-based methods of Neural Network, Graph Theoretic, Fuzzy and Cluster have, to some extent, been effective. text summary.Both-out and short-term modes have been researched. Many summing methods are based on extraction methods. The Abstractive method is similar to man-made abbreviations. Incomprehensible summaries currently require heavy equipment for language production and are difficult to replicate in certain areas of the domain.

10.REFERENCES:

- [1] G. Vijay Kumar, Aravind Yadav, B.Vishnu Priya,M.Naga Lahari, J.SmritiD.Samved Reddy,"Text Summarization Using NLP"CC BY-NC 4.0.
- [2] Neelima Bhatia and Arunimal Jaiswal,"Automatic Text Summarization and itsMethods-AReview", 6th International Conference.
- [3] Deepali K.Gaikwad, C.Namrata Mahender,"A Review On TextSummarization", "International Journal of Adavanced Research in Computer andCommunication Engineering", Vol.5, Issue 3, March 2016.
- [4] Pratibha Devishour, Naseer R, "Automatic Text Summarization Using Natural Language Processing", International research journal of Engineering and Technology (IRJET) Volume.
- [5] Aditya Jain, Divij Bhatia, Manish K Thakur's "Extractive TextSummarization using Word Vector Embedding"(2017).
- [6] Paul Gigioli, Nikhita Sagar, Anand Rao, Joseph Voyles "Domain-Aware Abstractive Text Summarization for MedicalDocuments" published during 2018 IEEE BIBM.

- [7] A.T. Al-Taani, "Automatic text summarization approaches," 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017, pp.93-94, doi: 10.1109/ICTUS.2017.8285983.
- [8] A. P. Patil, S. Dalmia, S. Abu Ayub Ansari, T. Aul and V. Bhatnagar, "Automatic text summarizer," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014, pp. 1530-1534, doi:10.1109/ICACCI.2014.6968629.
- [9] I. Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand and P.K. Soni, "Natural Language Processing (NLP) based Text Summarization - A Survey," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp.13101317, doi:10.1109/ICICT50816.2021.9358703.