**Natural Language Processing for Text Summarization**

**Introduction:**

Natural Language Processing (NLP) revolutionizes text summarization, condensing information for myriad applications. It encompasses two main methods: extractive, which selects key sentences, and abstractive, which paraphrases content. Recent NLP advancements, notably large-scale language models, bolster abstractive summarization's quality. NLP-driven summarization plays a pivotal role in information retrieval, document analysis, and content generation across domains. Extractive techniques streamline by directly selecting essential data, while abstractive methods strive for human-like summaries through rephrasing and synthesis. As NLP evolves, its impact on text summarization expands, offering improved efficiency and accuracy in distilling large volumes of text into concise and informative summaries, vital for managing and understanding vast textual information.

**Abstract:**

Natural Language Processing (NLP) drives text summarization, condensing information for diverse applications. Methods encompass extractive, selecting vital sentences, and abstractive, paraphrasing content. Recent strides in NLP, notably large-scale language models, bolster abstractive summarization efficacy. NLP-driven summarization plays a pivotal role in information retrieval, document analysis, and content generation across various domains. Extractive techniques simplify the process by selecting salient information directly from the source, while abstractive methods strive for human-like summaries by rephrasing and synthesizing. As NLP continues to evolve, its impact on text summarization expands, offering improved efficiency and accuracy in distilling large volumes of textual data into concise and informative summaries**.**

**Description:**

Natural Language Processing (NLP) for text summarization involves leveraging computational techniques to condense textual information while retaining its essential meaning. It encompasses various methodologies, including extractive and abstractive summarization. Extractive methods identify and select key sentences or phrases directly from the original text, while abstractive approaches involve generating new phrases or sentences that convey the main ideas in a condensed form.

NLP-based text summarization finds applications in a wide range of domains, such as news aggregation, document summarization, and content generation. Recent advancements in NLP, particularly with the advent of large-scale pre-trained language models, have significantly improved the quality and effectiveness of text summarization systems.

Overall, NLP-driven text summarization serves as a vital tool for efficiently processing and understanding large volumes of textual data, enabling users to glean insights and information from complex documents more effectively.

**Program:**

importnltk

nltk.download('punkt')

nltk.download('stopwords')

fromnltk.tokenizeimportsent\_tokenize, word\_tokenize

fromnltk.corpusimportstopwords

fromnltk.probabilityimportFreqDist

deftext\_summarization(text, num\_sentences):

    # Tokenize the text into sentences

    sentences = sent\_tokenize(text)

    # Tokenize each sentence into words

    words = word\_tokenize(text.lower())

    # Remove stop words

    stop\_words = set(stopwords.words('english'))

    words = [word for word in words if word notinstop\_words]

    # Calculate word frequency

    word\_freq = FreqDist(words)

    sentence\_scores = {}

    for sentence in sentences:

        for word inword\_tokenize(sentence.lower()):

            if word inword\_freq:

                if sentence notinsentence\_scores:

                    sentence\_scores[sentence] = word\_freq[word]

                else:

                    sentence\_scores[sentence] += word\_freq[word]

    # Sort sentences by score and select top n sentences for summary

    top\_sentences = sorted(sentence\_scores.items(), key=lambda x: x[1], reverse=True)[:num\_sentences]

    summary = ' '.join([sentence[0] for sentence intop\_sentences])

    return summary

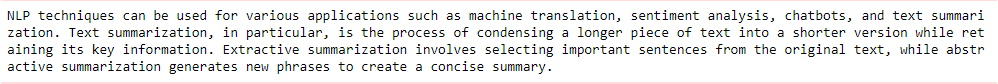
text = """

num\_sentences = 3  # Number of sentences in the summary

summary = text\_summarization(text, num\_sentences)

print(summary

**Output:**



**Conclusion:**

In conclusion, Natural Language Processing (NLP) has profoundly impacted text summarization, offering methods to distill vast amounts of information into concise summaries. Both extractive and abstractive approaches have seen advancements, with extractive methods focusing on selecting key sentences and abstractive methods generating human-like summaries through paraphrasing. Recent progress, particularly with large-scale language models, has improved summarization quality. NLP-driven summarization finds applications in various domains, from news aggregation to document analysis. As NLP continues to evolve, its role in text summarization will likely expand, offering more efficient and accurate ways to process and understand textual data, ultimately enhancing information retrieval and comprehension**.**