TEXT SUMMARIZATION SYSTEM

A PROJECT REPORT

Submitted by

PREMKUMAR.V (2303811714821032)

in partial fulfillment of requirements for the award of the course

AGI1242 – MACHINE LEARNING TECHNIQUES

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"TEXT SUMMARIZATION SYSTEM"** is the bonafide work of **PREMKUMAR.V (2303811714821032),** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr.T.AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram–621112.

SIGNATURE
Mr.R.ROSHAN JOSHUA.,M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K. Ramakrishnan College of

Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on …7/12/2024…

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“TEXT SUMARIZATION SYSTEM”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the award of the course **AGI1242 – MACHINE LEARNING TECHNIQUES.**

Signature

V. Premkumar

PREMKUMAR.V

Place: Samayapuram

Date: 5/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, "**K. Ramakrishnan College of Technology (Autonomous)**",for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,**Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN M.E., Ph.D.,** Head of the Departmentof **ARTIFICIAL INTELLIGENCE**, for providing his encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mr.R.ROSHAN JOSHUA., M.E.,** Department of **ARTIFICIAL INTELLIGENCE,** for his incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education.

MISSION OF THE INSTITUTION

Produce smart technocrats with empirical knowledge who can surmount the global challenges.

Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students.

Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations.

VISION OF DEPARTMENT

To become a renowned hub for AIML technologies to producing highly talented globally recognizabletechnocrats to meet industrial needs and societal expectation.

MISSION OF DEPARTMENT

Mission 1: To impart advanced education in AI and Machine Learning, built upon a foundation in Computer Science and Engineering.

Mission 2: To foster Experiential learning equips students with engineering skills to tackle real-worldproblems.

Mission 3: To promote collaborative innovation in AI, machine learning, and related research and development with industries.

Mission 4: To provide an enjoyable environment for pursuing excellence while upholding strong personal and professional values and ethics.

PROGRAM EDUCATIONAL OBJECTIVES

Graduates will be able to:

1. PEO1: Excel intechnical abilities tobuild intelligent systems in thefields of AI & ML in order tofind new opportunities

2. PEO2: Embrace new technology to solve real-world problems, whether alone or as a team, while prioritizing ethics and societal benefits.

3. PEO3: Accept lifelong learning to expandfutureopportunities inresearch and product development.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

Environment and sustainability: Understand the impact of the professional engineering solutionsin societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

This project aims to develop an automated text summarization system that generates concise summaries from lengthy texts. The system will utilize both extractive and abstractive techniques to condense information: extractivesummarization selects key sentences based on importance using methods like TF-IDF or TextRank, while abstractive summarization generates new, concise versions of the text through advanced deep learning models suchas Seq2Seq or transformer-based architectures like BERT and GPT. The goal is to create a tool that can be used across various applications, including news aggregation, content summarization, and research, offering users a wayto quickly grasp the core ideas of long texts without losing critical information..

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

NLP	-	Natural Language Processing
TF-IDF	-	Term Frequency-Inverse Document Frequency
Seq2Seq	-	Sequence-to-Sequence
BERT	-	Bidirectional Encoder Representations from Transformer
GPT	-	Generative Pre-trained Transformer
GUI	-	Graphical User Interface
API	-	Application Programming Interface
LSTM	-	Long Short-Term Memory
GRU	-	Gated Recurrent Unit
AI	-	Artificial Intelligence
DL	-	Deep Learning

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

In today's information-driven world, the ability to quickly and efficiently process vast amounts of textual data is crucial. With an overwhelming influx of contentfrom various sources, it becomes increasingly challenging to extract relevant insights in a timely manner. Text summarization, a branch of Natural Language Processing (NLP), addresses this challenge by generating shorter, more concise versions of long texts while preserving their key ideas and information.

This project focuses on developing a text summarization system that uses both extractive and abstractive methods to condense text effectively. Extractive summarization works by identifying and selecting key sentences directly from the source text based on their importance, while abstractive summarization generates new sentences that summarize the original content, often requiring advanced models like Seq2Seq or transformers. The system will be designed to handle a variety of applications, from news aggregation and academic research to summarizing documentsfor better accessibility.

The primary goal of this project is to build a reliable tool that enables users to understand the essence of large documents quickly, without losing important details. By incorporating state-of-the-art techniques in machine learning and deep learning, the system will improve over time, offering more accurate and contextually relevantsummaries.

1.2 PURPOSE AND IMPORTANCE OF THE PROJECT

The primary purpose of this project is to develop a text summarization system that simplifies large volumes of textual data into concise, meaningful summaries. As theworld generates an overwhelming amount of information daily, individuals and organizations face challenges in efficiently processing and understanding extensive content. A robust summarization tool can save time, enhance decision-making, and improve access to critical information.

The importance of this project lies in its wide-ranging applications across various domains:

- **News Media**: Quickly summarizing breaking news for better accessibility.
- **Education**: Assisting students and researchers by condensing academic papers or books.
- **E-commerce**: Summarizing product reviews to provide users with a clear consensus.
- **Healthcare**: Extracting key information from lengthy medical reports.
- **Business and Law**: Creating concise briefs from extensive legal or corporate document.

1.3 OBJECTIVES

The primary objective of the text summarization system is to develop an automated tool that condenses large volumes of text into concise, meaningful summaries.This overarching goal can be broken down into specific objectives:

1. **Develop a Hybrid Summarization Approach**
   - Implement both extractive and abstractive summarization techniques to handle diverse types of input text.
   - Optimize extractive methods to identify and compile key sentences effectively.
   - Use deep learning models to generate high-quality, human-like abstractive summaries.
2. **Enhance Efficiency and Accuracy**
   - Ensure the generated summaries retain the core meaning and context of the original text.
   - Address issues like redundancy, grammatical correctness, and semantic coherence.
3. **Adaptability for Multiple Domains**
   - Design a flexible system capable of summarizing various types of content, such as news articles, academic papers, legal documents, and customer reviews.
   - Enable customization for domain-specific summarization requirements.
4. **User-Friendly Implementation**
   - Provide an easy-to-use interface or API for end-users to input text and obtain summaries.
   - Ensure fast processing time for real-time or batch summarization tasks.
5. **Scalability and Automation**
   - Build a scalable system that can handle large datasets and work in cloud or enterprise environments.

- Enable integration with other systems for automated summarization workflows.

6. **Evaluation and Benchmarking**
    - Evaluate the system's performance using standard datasets like CNN/DailyMail and XSum.
    - Benchmark the system against state-of-the-art summarization tools

1.4 PROJECT SUMMARIZATION

This project focuses on developing an automated text summarization system to condense lengthy textual content into concise, meaningful summaries. Leveraging both extractive and abstractive summarization techniques, the system aims to provide accurate, contextually relevant, and grammatically correct summaries for various domains such as news, academic research, customer reviews, and legal documents.

The system will use traditional algorithms like TextRank for extractive summarization and advanced transformer-based models like BERT and GPT for abstractive summarization. Datasets such as CNN/DailyMail and XSum will be employedto train andevaluate the models. The project also emphasizes user-friendliness, scalability, and the ability to adapt to domain-specific requirements.

The outcome of this project is a robust and efficient summarization tool that saves time, enhances decision-making, and facilitates better information processing for individuals and organizations across diverse sectors.

CHAPTER 2

PROJECTMETHODOLOGY

The development of the text summarization system follows a structured approach that ensures clarity, efficiency, and effectiveness. This methodology is divided into key stages, each essential for building a robust and functional summarization system.

2.1 Problem Understanding and Requirement Analysis

The first step is to clearly define the problem and gather project requirements. Understanding the need for text summarization is crucial, as it will be applied across various domains such as news, education, legal, and e-commerce. The type of summarization technique is determined during this phase. The system will incorporate extractive summarization, which selects and combines key sentences directly from the input text, and abstractive summarization, which generates new sentences to summarizethe text more meaningfully. Furthermore, evaluation metrics like ROUGE (Recall- Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy) will be defined to measure the quality of the summaries generated.

2.2 Data Collection and Preprocessing

Data collection is essential to training and testing the summarization system. For this, publicly available datasets like CNN/DailyMail, XSum, and custom datasets specificto domains like e-commerce or legal documents will be used. The collected data will then becleaned to remove any noise, such as HTML tags, special characters, and redundant text.It is important to ensure that the data is well-labeled, with input texts and corresponding summaries. Preprocessing steps include normalizing the text (lowercasing), removing stopwords, and applying techniques such as tokenization and lemmatization. After preprocessing, the data will be split into training, validation, and testing sets to ensure proper model evaluation.

2.3 Design and Development

In this stage, the summarization techniques will be implemented. For extractive summarization, algorithms like TextRank or Latent Semantic Analysis (LSA) will be used to rank sentences based on importance. This will involve scoring sentences using techniques like Term Frequency-Inverse Document Frequency (TF-IDF) or cosine similarity. For abstractive summarization, transformer-based models such as BERT, GPT, or T5 will be fine-tuned for sequence-to-sequence tasks. These models use attention mechanisms to understand and generate text that reflects the original meaning. If a hybrid approach is necessary, combining extractive and abstractive summarization will be explored to balance the benefits of both methods.

2.4 Implementation

The implementation phase involves actual programming and system development. Python will be the primary language for programming, and libraries like TensorFlow, PyTorch, Hugging Face Transformers, and Scikit-learn will be used to develop and train the models. The summarization pipeline will be built, with attention paid to optimizing the system's performance, memory usage, and speed. An easy-to-use interface or API willalso be developed, enabling users to input text and receive summaries seamlessly.

2.5 Evaluation and Testing

After developing the system, it will be evaluated using various metrics to assess its effectiveness. The primary metrics used will be ROUGE and BLEU, which measure the overlap between the generated summary and the reference summary, as well as the quality of the language. In addition to automated evaluation, user feedback will be gatheredthrough usability tests to evaluate how readable, coherent, and useful the generated summaries are. This phase ensures that the system meets user expectations and provides high-quality summaries.

2.6 Deployment

The next step is deploying the summarization system to ensure it can be used in real - world scenarios. The system will be made accessible through web-based applications or APIs, allowing easy integration with other platforms. Cloud deployment on platforms likeAWS, Azure, or Google Cloud will be considered to ensure scalability and availability. Performance optimization will be carried out to handle large-scale text inputs and generate summaries in real-time.

2.7 Maintenance and Enhancement

Once the system is deployed, it will undergo continuous monitoring and maintenance. This phase ensures that the system remains effective over time by addressing issues such as loss of context or redundancy in summaries. Additionally, the system will beupdated regularly with new data to improve its accuracy and adapt to evolving requirements. Further enhancements, such as adding multi-document summarization and support for multiple languages, will be explored as the system matures.

2.8 DETAILED SYSTEM ARCHITECTURE DIAGRAM

The system architecture diagram depicts the overall structure of the processing tool, showcasing its major components and their interactions. diagram is divided into several sections, each representing a key aspect of system's functionality.

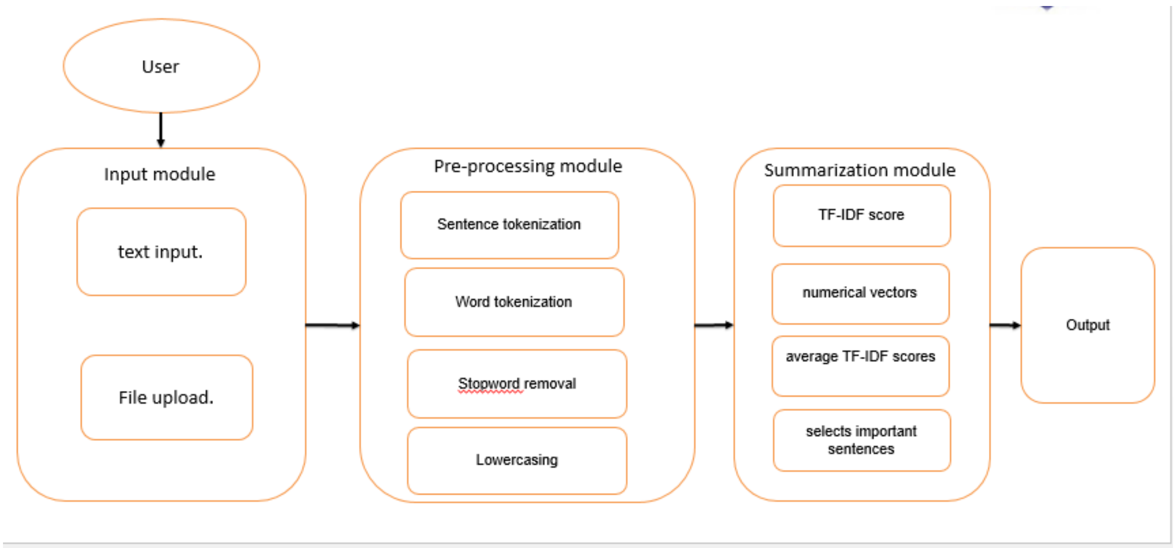


Fig 2.8.1 : Architecture Diagram

CHAPTER 3

MACHINE LEARNING PREFERANCE

In the context of text summarization, machine learning plays a critical role in automatically generating concise and meaningful summaries from large amounts of text. For this project, we aim to leverage machine learning techniques that are both effective and scalable. The choice of machine learning methods will be guided by the nature of the summarization task—whether it is extractive or abstractive—and the specific requirements of the system.

3.1 Extractive Summarization

For extractive summarization, where the goal is to select key sentences directly from the input text, the preferred machine learning techniques involve models that can effectively evaluate sentence importance and relevance.

- **TextRank**: This graph-based algorithm is one of the simplest yet effective techniques for extractive summarization. It ranks sentences based on their relevance and relationships to other sentences, similar to how Google's PageRank works for ranking web pages.
- **Supervised Learning with Classification Models**: Models like Logistic Regression, Support Vector Machines (SVM), or Random Forests can betrainedto classify sentences as either "important" or "not important." These model share trained on labeled datasets, where sentences are marked for their relevance to the summary.
- **Deep Learning (RNNs, LSTMs, GRUs)**: Recurrent Neural Networks (RNNs), especially with Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU), can be trained to learn sentence importance by processing the text sequentially. These models can capture dependencies over long text sequences, which is beneficial for understanding the context of each sentence within the

document.

3.2 Abstractive Summarization

Abstractive summarization involves generating entirely new sentences that capture the meaning of the original text. This requires more complex machine learning models, typically relying on sequence-to-sequence learning and transformer-based architectures.

- **Transformers (BERT, GPT, T5)**: Transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pretrained Transformer), and T5 (Text-to-Text Transfer Transformer), are ideal for abstractive summarization. These models use attentionmechanisms to understand the relationship between different words in a sentence, allowing them to generate fluent, coherent, and contextually accurate summaries. T5, in particular, is designed to handle a variety of NLP tasks, including summarization, by framing them as text- to-text problems.
- **Sequence-to-Sequence Models (Seq2Seq)**: These models consist of an encoder that processes the input text and a decoder that generates the summary. Sequence-to-sequence models, often enhanced with attention mechanisms, can be trained to output summaries that paraphrase the original content effectively. They are particularly useful for generating abstractive summaries that are both succinct and meaningful.
- **Pretrained Language Models**: Pretrained models like GPT-3 or BART (Bidirectional and Auto-Regressive Transformers) are highly effective for abstractive summarization due to their large-scale training on diverse text corpora. These models have shown remarkable capabilities in generating human-like text, which is crucial for creating fluent and accurate summaries.

3.3 Hybrid Models

In many cases, a hybrid approach combining both extractive and abstractive methods is beneficial. In this approach, an extractive model first identifies key sentences or sections from the text, and an abstractive model then generates a summary based on the selected content. This approach leverages the strengths of both methods to produce summaries that are both informative and coherent.

For instance, combining TextRank or TF-IDF-based methods with transformermodels like T5 could yield high-quality results. The extractive phase helps ensure the keypoints are captured, while the abstractive phase ensures the final summaryis fluent and paraphrased effectively.

3.4 Evaluation Metrics for Machine Learning Models

To assess the performance of the summarization models, we will use a range of evaluation metrics:

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: Measures the overlap between the generated summary and a reference summary, focusing on recall, precision, and F1 score.
- **BLEU (Bilingual Evaluation Understudy)**: Used to evaluate the fluency and accuracy of the generated summaries by comparing n-grams between the predicted summary and the reference.
- **Perplexity**: Often used for language models, perplexity evaluates the quality of generated text by measuring how well the model predicts the next word in a sequence.

3.5 Tools and Frameworks

To implement these machine learning techniques, we will rely on several powerful libraries and frameworks:

- **Hugging Face Transformers**: Provides easy access to pre-trained models like BERT, GPT, and T5, making it easier to fine-tune these models for summarization tasks.
- **TensorFlow/PyTorch**: Both frameworks are popular for building and training deep learning models, especially when working with large datasets and neural networks.

- **Scikit-learn**: Useful for traditional machine learning methods like TextRank or SVM-based approaches for extractive summarization.

CHAPTER – 4
MACHINE LEARNING METHODOLOGY

4.1 Defining the Problem Type

The first step in building a text summarization system is to define the type of summarization to be performed. There are two primary approaches: extractive summarization and abstractive summarization. Extractive summarization selects a subset of sentences from the original text that are most relevant to the summary. On the other hand, abstractive summarization generates new sentences that convey the key points of the text, often by paraphrasing or rewriting sections. For this methodology, we focus on extractive summarization, which uses unsupervised learning techniques like TF-IDF.

4.2 Data Collection and Preparation

Once the problem type is defined, the next step is to collect and prepare the data. This involves gathering a large corpus of text, which may include articles, papers, or other textual content. Some datasets also provide human-written summaries for supervised learning tasks. Preprocessing of the text data is essential for cleaning and preparing it for the machine learning model. This typically includes tokenizing the text into sentences and words, cleaning the text by removing special characters, punctuation, and stopwords, and normalizing it by converting all text to lowercase and handling special characters.

4.3 Feature Engineering

In this step, the raw text data is transformed into numerical features that machinelearning algorithms can process. The features are extracted to capture relevant information from the text. For extractive summarization, common methods for feature extraction include TF-IDF vectorization and Bag-of-Words (BoW), which are used to transform the text into numerical vectors. Additional textual features may include sentence length, sentence position, word frequency, and named entity recognition (NER). For instance, longer sentences might contain more information, and sentences located at the beginning of a document might be considered more relevant. Named entities, such as people, organizatio ns, and locations, are often significant and can be used to enhance the feature set.

4.4 Modeling

The modeling phase is where the core machine learning techniques are applied to the features to generate the summary. For extractive summarization, several unsupervised learning techniques can be used. One common method is to compute the TF-IDF score for each word in a sentence and then calculate the average score for the entire sentence. Sentences with higher scores are ranked higher, and the top-ranked sentences are selected for the summary. Another approach involves using clustering-based methods like K- means clustering or Latent Semantic Analysis (LSA), where similar sentences are grouped together based on the feature vectors. From each group, the most representative sentence is selected. Additionally, graph-based methods such as TextRank or LexRank are employed, where sentences are represented as nodes in a graph, and edges represent similarities between sentences. These methods rank sentences based on their importance inthe graph, with more central sentences being considered more relevant.

4.5 Model Evaluation

Once the model is trained, the next step is evaluating its performance. There are two types of evaluation: intrinsic and extrinsic. Intrinsic evaluation involves assessing the quality of the generated summaries by comparing them to human-written summaries. Common evaluation metrics include ROUGE scores, which measure the overlap of n-grams between the generated summary and reference summaries, and precision, recall, and F1-score, which measure how much relevant information is retrieved and how much is lost during summarization. Compression ratio, the ratio of the original document length to the summary length, can also provide insights into how much text was reduced to create the summary. Extrinsic evaluation, on the other hand, measures the real-world performance of the summaries. For example, how well the summaries support decision-making, information retrieval, or other tasks, often by gathering user feedback or conducting task-based evaluations.

4.6 Implementation

In this stage, the model is implemented to produce summaries from raw text. The process begins by preprocessing the input text, which includes tokenization, stopword removal, and normalization. Next, features are extracted from the text using methods like TF-IDF or word embeddings (e.g., Word2Vec, BERT), which provide more semantic depth to the features. Sentences are ranked based on their relevance scores, which are typically calculated using the TF-IDF values or other feature-based ranking methods. Finally, the top-ranked sentences are selected to form the summary, ensuring that the most relevant and informative parts of the original text are preserved in the summary.

CHAPTER -5

MODULES

5.1 Input Module

This module handles receiving text data from the user. It accepts either direct input (raw text) or uploaded files in various formats, ensuring the text is properly captured for processing.

5.2 Preprocessing Module

The preprocessing module cleans and organizes the input text. This involves tokenization (splitting text into sentences or words), removal of stopwords (common words like "the" or "is"), text normalization (converting text to lowercase and removing special characters), and optionally stemming or lemmatization (reducing words to their root forms).

5.3 Feature Extraction Module

This module extracts key features from the text that are crucial for summarization. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency), sentence position analysis, and named entity recognition are used to evaluatethe importance of sentences within the document.

5.4 Summarization Model Module

The core of the system lies here. For extractive summarization, it ranks sentences based on importance and selects the top ones for the summary. For abstractivesummarization, advanced models like transformers (BERT, GPT, T5) are used to generatehuman-like, rephrased summaries.

5.5 Evaluation Module

The evaluation module ensures the quality of the generated summary by using metrics such as ROUGE**,** precision**,** recall**,** and F1-score**.** It also assesses how well the summary retains the key information compared to the original text.

5.6 Output Module

This module presents the summary to the user. It can display the summary as texton the screen or allow it to be downloaded as a file. Users might also have options to customize the summary's length or content.

5.7 Advanced Features Module

Optional modules include user feedback**,** which collects user opinions to improvethe system, and data augmentation, which expands the dataset by generating synthetic summaries for training purposes.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

2.1 CONCLUSION

In conclusion, developing a Text Summarization System involves a series of well-defined modules that work together to process raw text and produce concise, relevant summaries. The system begins by receiving and preprocessing the input text, which includes tokenization, stopword removal, and normalization. The next step involves extracting meaningful features such as TF-IDF scores, sentence position, and named entities, which provide the system with the necessary information to determine the importance of each sentence. The core of the summarization system lies in the summarization model module, where techniques like extractive summarization (using TF-IDF or TextRank) or more advanced abstractive summarization methods (such as Transformer models) are employed. Once the summary is generated, it is evaluated using metrics like ROUGE, precision, recall, and F1-score, ensuring the quality and relevance of the content. The output module then presents the summary to the user, either on-screen or as a downloadable file, offering a seamless user experience. Optional advanced features, such as user feedback and data augmentation, further enhance the system's performance and adaptability over time. Ultimately, a well-designed text summarization system can significantly improve information retrieval and decision-making processes, saving time and effort by extracting the most important information from large volumes of text. With advancements in machine learning and natural language processing, such systems can continue to evolve and provide even more accurate, context-aware summaries for various applications.

2.2 Future Scope of Text Summarization Systems

1. Improved Abstractive Summarization

Future advancements will enhance abstractive summarization, allowing models to generate more accurate, human-like summaries. Deep learning techniques, particularly transformers, will continue to improve, enabling models tounderstand and rephrase content with greater fluency.

2. Multilingual Summarization

With globalization, there is a growing need for multilingual summarization. Future systems will be able to summarize content across multiple languages, making summaries accessible to a broader audience and enhancing cross-lingual communication.

3. Context-Aware Summarization

Context-aware summarization will allow models to better understand the intent and context behind the text, tailoring summaries to suit the genre, audience, and purpose, providing more accurate and relevant outputs.

4. Personalized Summarization

Future systems will leverage personalization, adjusting summaries based on individual user preferences, interests, and past behavior, ensuring that the summary is highly relevant and tailored to each user's needs.

5. Integration with Knowledge Graphs

By integrating knowledge graphs and external databases, future summarization models will improve their understanding of entities and concepts, producing summaries that are more informative and contextually enriched.

6. Real-Time Summarization

As demand for instant information grows, real-time summarization systems will allow users to receive immediate summaries of live data, such as news articles, social media posts, or live broadcasts, helping to stay updated without the need toread through entire texts.

7. Ethical and Fair Summarization

Future models will focus on reducing bias and ensuring fairness in summarization. Ethical considerations will be integrated into the design of models, ensuring that summaries are accurate, diverse, and unbiased.

8. Interactive and Multimodal Summarization

Advancements in interactive summarization will allow users to adjust summaries dynamically. Additionally, multimodal summarization, which combines text with images, videos, and other media, will offer richer, more comprehensive summaries.

9. Scalability and Efficiency

Improved scalability and efficiency will ensure that summarization systems can handle vast amounts of data quickly and with minimal resources, making themmore practical for large-scale applications.

10. Integration with Other NLP Tasks

Future summarization systems will integrate seamlessly with other NLP tasks like sentiment analysis and topic modeling, producing summaries that are not only informative but also contextually aware of the emotional tone and key themes.

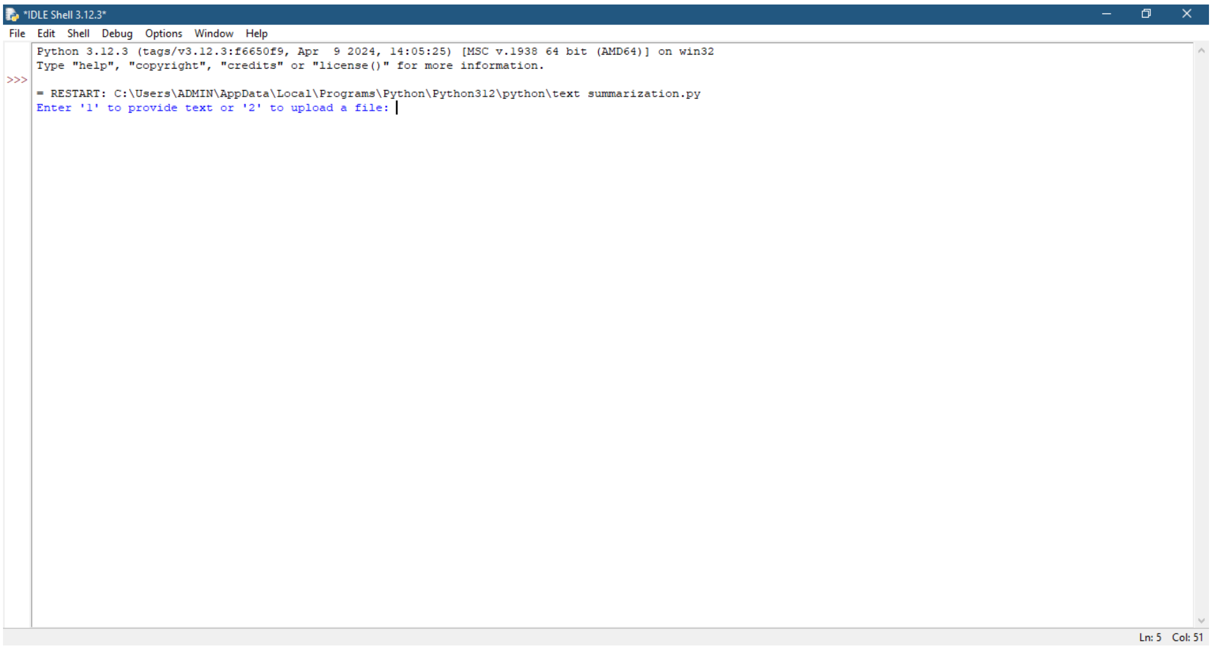
APPENDICES

APPENDIX A-SOURCE CODE

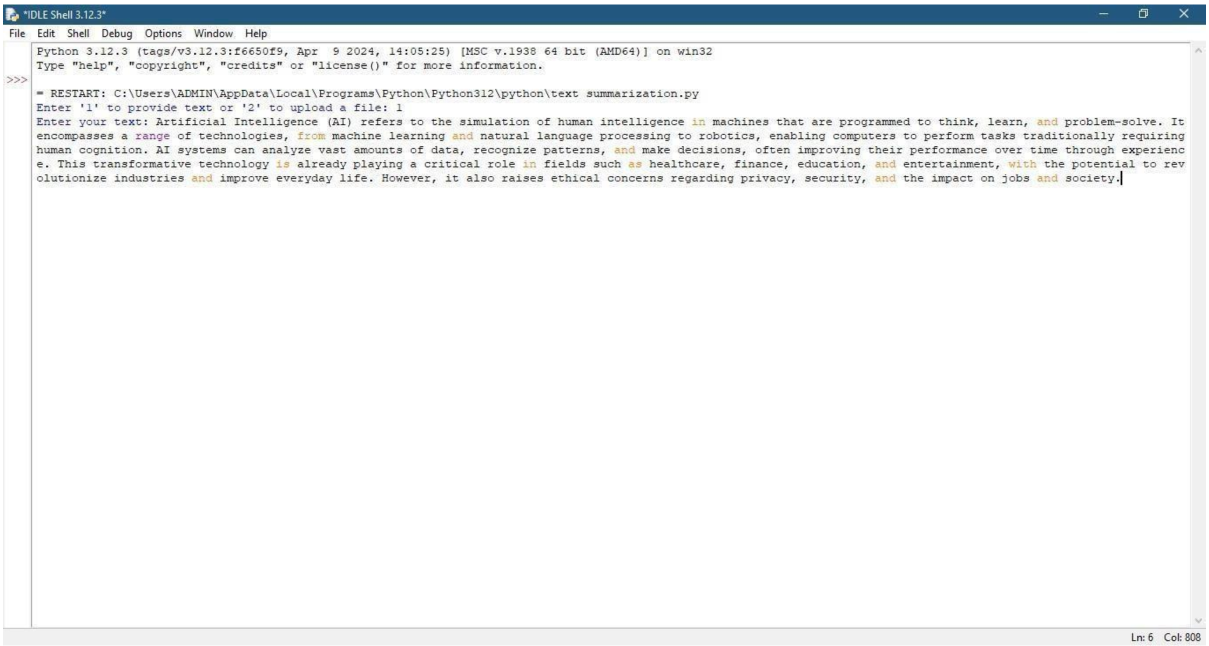
```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
import numpy as np

# Download NLTK data
nltk.download('punkt')
nltk.download('stopwords')

def extractive_summary(text):
    """
    Generate an extractive summary from the input text by selecting important sentences.
    """
    # Tokenize text into sentences
    sentences = sent_tokenize(text)

    # Preprocess text: Tokenize, remove stopwords, and lowercase
    stop_words = set(stopwords.words("english"))
    preprocessed_sentences = [
        " ".join(
            word for word in word_tokenize(sentence.lower())
            if word.isalnum() and word not in stop_words
        )
        for sentence in sentences
    ]

    # Compute TF-IDF scores
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(preprocessed_sentences)
    sentence_scores = np.mean(tfidf_matrix.toarray(), axis=1)

    # Determine the number of important sentences (e.g., 30% or at least 5)
    num_sentences = max(5, int(len(sentences) * 0.3))

    # Rank sentences based on scores
    ranked_indices = np.argsort(sentence_scores)[-num_sentences:]
    ranked_indices = sorted(ranked_indices) # Preserve original sentence order

    # Extract important sentences
    important_sentences = [sentences[i] for i in ranked_indices]
    return important_sentences

def get_input():
    """
    Prompt the user to provide text directly or upload a text file.
    """
    choice = input("Enter  '1' to provide text or  '2' to upload a  file: ").strip()
    if choice == '1':
        text = input("Enter your text: ")
    elif choice == '2':
        file_path = input("Enter the file path: ").strip()
        try:
            with open(file_path, 'r', encoding='utf-8') as file:
                text = file.read()
        except FileNotFoundError:
            print("File not found. Please check the file path.")
            return None
    else:
        print("Invalid choice. Please enter '1' or '2'.")
        return None
    return text

if__name__== "__main__":
    # Get input from the user
    user_text = get_input()

    if user_text:
        # Generate a summary with important sentences
        important_sentences = extractive_summary(user_text)

        # Display the summary as important sentences
        print("\nSummary:")
        for idx, sentence in enumerate(important_sentences, 1):
            print(f"{idx}. {sentence}")
```

APPENDIX B - SCREENSHOTS
RESULT AND DISCUSSION

INPUT:

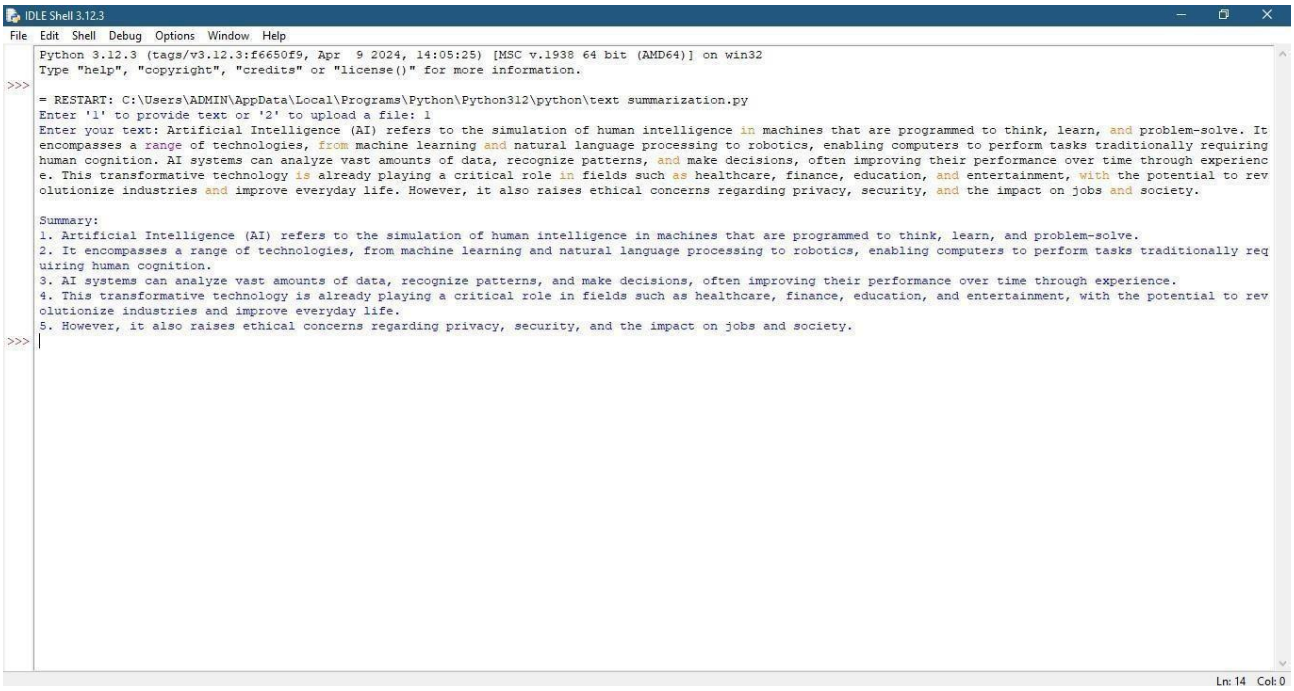


```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ADMIN\AppData\Local\Programs\Python\Python312\python\text summarization.py
Enter '1' to provide text or '2' to upload a file: 
```



```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ADMIN\AppData\Local\Programs\Python\Python312\python\text summarization.py
Enter '1' to provide text or '2' to upload a file: 1
Enter your text: Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and problem-solve. It
encompasses a range of technologies, from machine learning and natural language processing to robotics, enabling computers to perform tasks traditionally requiring
human cognition. AI systems can analyze vast amounts of data, recognize patterns, and make decisions, often improving their performance over time through experienc
e. This transformative technology is already playing a critical role in fields such as healthcare, finance, education, and entertainment, with the potential to rev
olutionize industries and improve everyday life. However, it also raises ethical concerns regarding privacy, security, and the impact on jobs and society.
```

OUTPUT:



```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ADMIN\AppData\Local\Programs\Python\Python312\python\text summarization.py
Enter '1' to provide text or '2' to upload a file: 1
Enter your text: Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and problem-solve. It
encompasses a range of technologies, from machine learning and natural language processing to robotics, enabling computers to perform tasks traditionally requiring
human cognition. AI systems can analyze vast amounts of data, recognize patterns, and make decisions, often improving their performance over time through experienc
e. This transformative technology is already playing a critical role in fields such as healthcare, finance, education, and entertainment, with the potential to rev
olutionize industries and improve everyday life. However, it also raises ethical concerns regarding privacy, security, and the impact on jobs and society.

Summary:
1. Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think, learn, and problem-solve.
2. It encompasses a range of technologies, from machine learning and natural language processing to robotics, enabling computers to perform tasks traditionally req
uiring human cognition.
3. AI systems can analyze vast amounts of data, recognize patterns, and make decisions, often improving their performance over time through experience.
4. This transformative technology is already playing a critical role in fields such as healthcare, finance, education, and entertainment, with the potential to rev
olutionize industries and improve everyday life.
5. However, it also raises ethical concerns regarding privacy, security, and the impact on jobs and society.
>>>
```