Noida institute of Engineering & Technology

NIET
Greater Noida
Estd.2001
Get Future Ready
Autonomous Institute

Department of CSE (AI)

Session (2024-25)

Programming for Data Analytics Project (ACSAI0617) Presentation on

“TEXT SUMMARIZATION USING PYTHON”

Project Guide:

Mrs. Garima Jain
Deputy Head, CSBS

Submitted To:

Mrs. Garima Jain

Group Members:

Shreya Vishwakarma(2201331520171)
Prasoon (2201331520130)
Parul Goel (2201331520124)

Presentation Outline

- Introduction
- Literature Review
- Research Gap
- How to overcome Research Gap
- Objectives of the Research work
- Methodology
- Result
- Conclusion
- References

Introduction

- Text summarization condenses large volumes of text into concise summaries, helping reduce information overload and enhance understanding.
- This system uses Python-based NLP and machine learning techniques to generate extractive or abstractive summaries.
- It analyzes input documents to identify key sentences or generate new concise representations.
- Aims to provide an efficient, automated, and scalable solution for summarizing text data from various sources.
- Helps users quickly grasp essential information, improving productivity and decision-making through intelligent content summarization.

Literature Review

Year	Paper Title	Author(s)	Research Gap
2016	TextRank: Bringing Order into Texts	Rada Mihalcea, Paul Tarau	Lacks semantic understanding; purely graph-based approach
2017	A Deep Reinforced Model for Abstractive Summarization	Romain Paulus, Caiming Xiong, Richard Socher	Limited ability to handle long documents
2019	Pre-training for Abstractive Text Summarization	Yang Liu, Mirella Lapata	Fine-tuning is task-specific; generalization is limited
2020	BERTSum: Fine-tuning BERT for Extractive Summarization	Yang Liu, Mirella Lapata	Focuses on extractive summaries only, not suitable for abstractive needs
2020	PEGASUS: Pre-training with Extracted Gap-sentences for Summarization	Jingqing Zhang et al.	High computational cost and dependency on large datasets
2021	A Survey on Abstractive Text Summarization	P. Yadav, M. Ekbal, P. Bhattacharyya	Lacks comprehensive evaluation metrics for summary quality

Research Gap

- **Limited Real-Time Data Integration:** Many existing summarization systems do not handle real-time text streams (e.g., news, chats), limiting their usability in dynamic environments.

- **Accessibility Barriers:** Some Python-based tools and frameworks are too technical, making them difficult to use for non-programmers or those without NLP expertise.

- **Lack of Personalization:** Current models often generate generic summaries and fail to adapt to user preferences (e.g., summary length, focus area, domain-specific content).

- **User Engagement Challenges:** Poorly designed interfaces or lack of interactivity in summarization tools can lead to reduced user interest and adoption.

How to Overcome the Research Gap

- **Real-Time Data Integration:** Integrate with live data sources such as news feeds, social media APIs, or real-time chats to enable up-to-date summarization.

- **User-Centric Design:** Develop an intuitive, multilingual user interface compatible with desktop and mobile platforms to ensure accessibility for users of all backgrounds.

- **Personalized Recommendations:** Customize summaries based on user preferences such as preferred length, tone (formal/informal), domain-specific focus (legal, medical, educational), and language.

- **Extensibility:** Design a modular architecture allowing for the future addition of features like sentiment-aware summarization, voice-to-summary conversion, and adaptive summarization based on user interaction patterns.

NIET Greater Noida
Get Future Ready Estd.2001 Autonomous Institute

Objective of the Research work

Primary Objective: Develop an intelligent, Python-based system that utilizes machine learning and NLP techniques to generate accurate and concise summaries from large text documents.

Supporting Goals:

- Collect high-quality datasets from diverse sources such as news articles, research papers, and blogs for training and evaluation.
- Train and compare multiple machine learning and deep learning models (e.g., TextRank, BERT, GPT) to determine the most effective summarization technique.
- Design a scalable, user-friendly web or mobile application that performs summarization efficiently, even on low-resource devices.
- Enhance user productivity by enabling quick understanding of lengthy content, supporting better decision-making and information processing.

Methodology

1. Collect text data from various trusted sources.

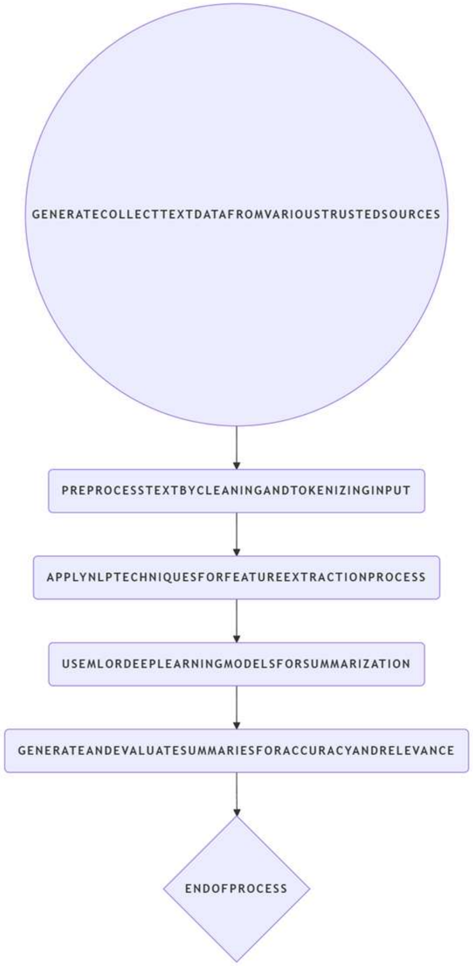
2. Preprocess text by cleaning and tokenizing input.

3. Apply NLP techniques for feature extraction process.

4. Use ML or deep learning models for summarization.

5. Generate and evaluate summaries for accuracy and relevance.

Methodology



GENERATECOLLECTTEXTDATAFROMVARIOUSTRUSTEDSOURCES

PREPROCESSTEXTBYCLEANINGANDTOKENIZINGINPUT

APPLYNLPTECHNIQUESFORFEATUREEXTRACTIONPROCESS

USEMLORDEEPLEARNINGMODELSFORSUMMARIZATION

GENERATEANDEVALUATESUMMARIESFORACCURACYANDRELEVANCE

ENDOFPROCESS

Result

Sample Outputs:

Original Text Word Count: 43

Summary Word Count: 11

User Testing:

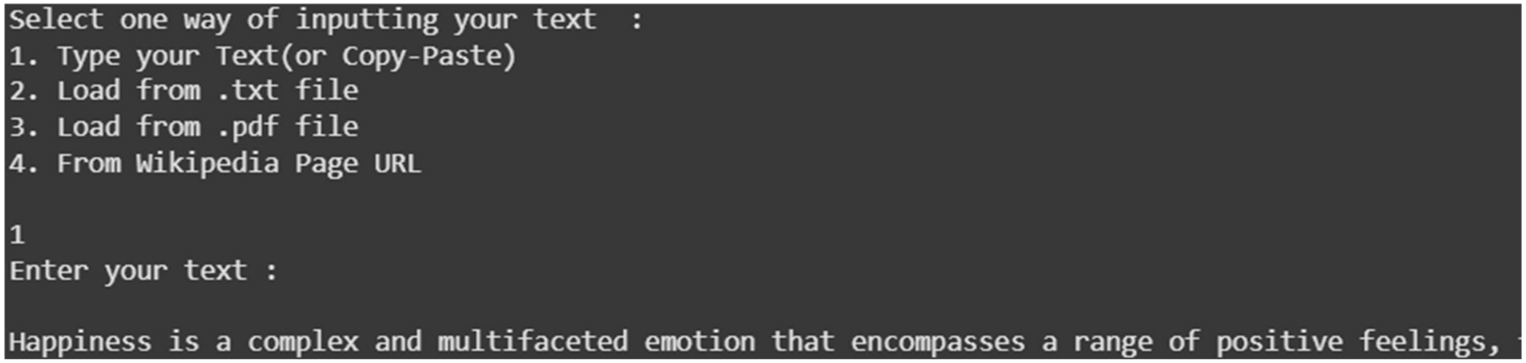
Compression Ratio: 74.41%

Impact:

Output: Clear, informative, and in original phrasing.

Result

Input:



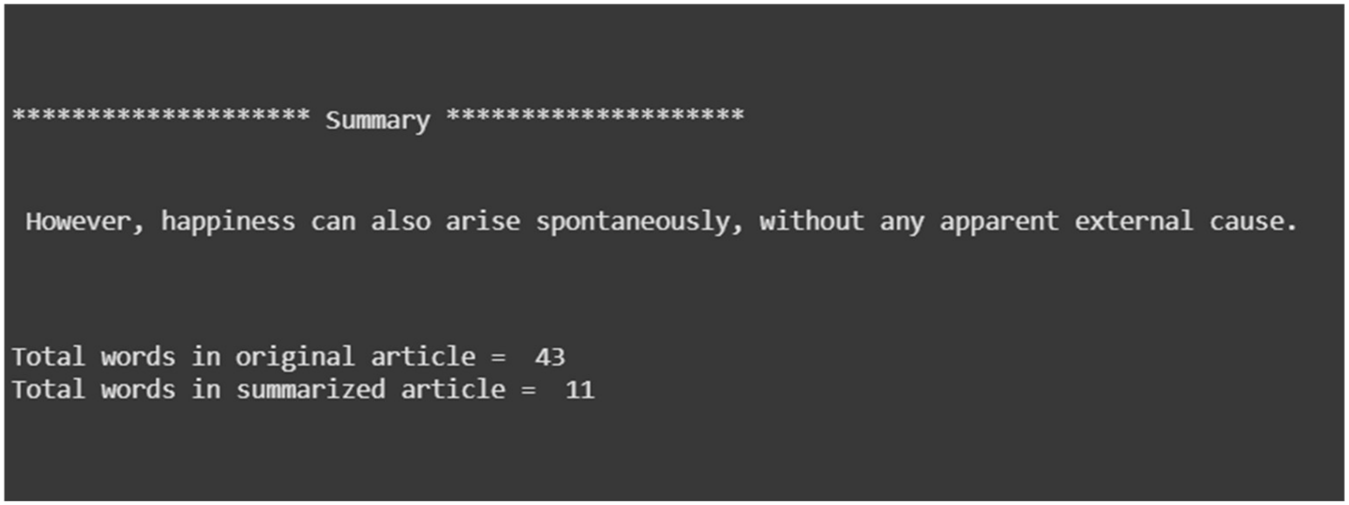
```
Select one way of inputting your text  :
1. Type your Text(or Copy-Paste)
2. Load from .txt file
3. Load from .pdf file
4. From Wikipedia Page URL

1
Enter your text :

Happiness is a complex and multifaceted emotion that encompasses a range of positive feelings,
```

Result

Output:



```
******************** Summary ********************

 However, happiness can also arise spontaneously, without any apparent external cause.


Total words in original article =  43
Total words in summarized article =  11
```

Conclusion

Summary:

The developed system provides automated, Python-based text summarization using natural language processing and machine learning techniques to extract key information from large texts efficiently.

Impact:

Enhances reading efficiency, aids quick understanding, reduces manual summarization effort, and supports decision-making in education, research, and business.

Future Enhancements:

Include support for abstractive models, integrate voice-to-text summarization, add multilingual support, and enable offline summarization for mobile applications.

Thank You.