Text Summarizer using NLP A innovative text summarizing application

Dr. Dayanand J[1], Anjali Sultanpure[2]

[1]*Professor, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka*

[2]*PG Student, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka*

***Abstract*—A summary is a short narrative of a long-written passage with the author's most important ideas. Summarizing helps develop your reading and writing skills. To sum up, you need to read a passage carefully, find the major and supporting ideas. One has to understand the difference between a summary and a paraphrase. A paraphrase is simply a rewriting of a passage in your own words. On the other hand, a summary contains only the main idea and the supporting ideas of a passage. A summary will be much shorter than a paraphrase.**

***Keywords*—Web; Machine Learning; Editors; React; Web Portal; Web Technology; NLP; Natural Language Processing; JavaScript**

I.INTRODUCTION

The revolution of the world of information through the means of internet has been more than ever before. The internet has brought soul to the IT world. This has set the global possibility for the unparallel level of capability integration. The information can be accessed from any geographic location.

A very active area of research right now is automatic text processing. Automatic summarization, which involves condensing a text while maintaining its information content, is a crucial task in this area. A summarizer is a system that provides users with a reduced version of its inputs. The entire goal of text summarization is to reduce the source text to a manageable size while maintaining the information's overall meaning and content. This model aids us by streamlining the enormous amount of information that people interact with on a daily basis, which improves and lowers the time complexity.

Summary construction is generally a challenging undertaking that ideally requires strong natural language processing skills. Right now research is concentrated on extractive-summary generation to simplify the issue. Simply put, an extractive summary is a condensed version of the original text. These summaries cannot provide adequate narrative consistency, but they can easily represent a rough idea of the text's substance for relevance assessment.

A summary can be used as an informational tool to cover all pertinent material in the text, or it can be used in a suggestive manner to point to certain passages in the original document. The shorter reading time of a summary is its primary benefit in both situations. The size of the summary may be modified, its content is determinist, and it is simple to establish the relationship between a text element's location in the summary and its place in the original text when a summary is generated automatically, among other benefits.

II.REQUIREMENT ANALYSIS

A. Software Requirement Specification

- Text summarization reduces the length of the original text while maintaining the information and overall meaning. It is very challenging for humans. Manual text summarization of lengthy documents. There are two types of text summarization techniques: extractive and abstractive. Selecting significant sentences, paragraphs, etc. from the original document and concatenating them into a shorter form constitutes an extractive summarization method
- However, The linguistic and statistical characteristics of sentences are used to determine their importance. Understanding the original text and retelling it in fewer words are the components of an abstractive summarization method.

- The text is examined and interpreted using linguistic techniques, and after that, new ideas and expressions are found to best describe it by creating a new, shorter text that highlights the most crucial details from the original text document. This paper presents a survey of text summarization and extraction techniques.

B. Purpose of the Project

The In the rapidly expanding information age of today, text summarization has emerged as a crucial and useful tool for aiding and interpreting text information. Humans find it extremely challenging to manually summaries lengthy texts. On the internet, there are a lot of text resources available.

On the other hand, the Internet typically offers more information than is required. There are two issues that arise as a result: searching through the vast number of documents that are available for relevant documents and taking in the vast amount of relevant information. The objective of automatic text summarization is to reduce the source text while retaining its informational content and overall meaning.

A summary can be used as an informative tool to cover all pertinent information in the text, or it can be used as a pointer to specific sections of the original document. The shorter reading time of a summary is its primary benefit in both situations. A good summary system should reflect the document's variety of topics while minimizing redundancy.

C.Intended Audience

Automatic text summarizing techniques that make it simple for users to form inferences from them were required due to the present growth of unstructured textual data in the digital domain. We now have instant access to a wealth of knowledge. However, the majority of this information is superfluous, unimportant, and might not be understood as intended. Summary construction is generally a challenging task that ideally requires strong natural language processing skills. Current research is concentrated on extractive-summary generation to simplify the issue. Simply put, an extractive summary is a condensed version of the original text. Although they cannot guarantee strong narrative coherence, these summaries can be used to quickly represent a text's general content when determining its relevance.

D.Application Functionality

The application mainly focuses on converting the original huge text to summarizing the same text. This has a huge requirement in the feedback world for example the online ecommerce product comments or feedback where the comments can be summarized to the so that the new user who browses through the product take less time to read the comment or the feedback. This saves time of the user and gets a complete understanding on the comments.

The following are the major functionalities prescribed for the project idea.

E.Application Features

The following are the application features which are mainly focused for the project idea.

- Portable: The proposed application can be consumed at any time on any environment irrespective of system software or hardware specification.
- Ready to Use: No installation or configuration is required.
- User friendly: User can easily operate the application as it is developed under the user experience consideration as the primary intent.
- Performance: User should be able to phrase the data at faster response time

F.System requirements

- Software Requirements:

1. Operating System: Linux Operating System
2. Frontend Technologies: ReactJS, NextJS
3. Frontend Libraries: Material UI
4. ML packages: Keras
5. Flask Framework
6. Nginx Server

- Hardware Requirements:

1. Processor with min. 1.2GHz Clock Speed
2. Min 512 MB RAM
3. 8GB Storage Space

III.SYSTEM DESIGN

The architecture for the proposed system requires a highly performing backend server with a minimum of 700 MHz clock speed. This server runs on the Linux based operating system which gives the facility to establish a cloud compiler server. Linux OS like

Ubuntu, Fedora etc. can be used. The greater the performance of the CPU greater the compilation speed. The design is majorly divided into two layers:

1. Frontend Layer
2. Backend Layer

The Layers of our system Layers is as shown in Fig. 1

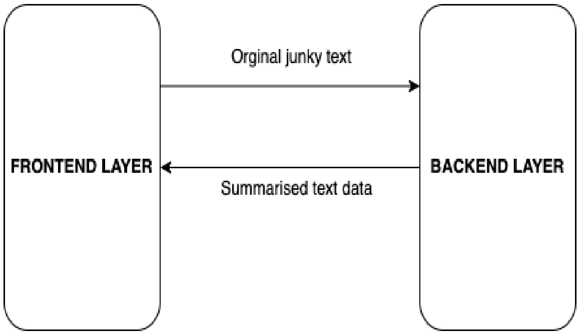


Fig 1: System Layers

G.Frontend Layer

The presentation layer of the system i.e., the user interface of the system is the playground for the user to give their inputs. This layer is responsible for taking the input from the user and display the summarized output once the processing is done.

H.Backend Layer

The backend here refers to the NodeJs server from where the compiler processes are spawned, and the execution is initiated. The Linux server with the compilers of various high level programming languages processes running, the NodeJs root module calls these processes and executes the program. The flow is as shown in Fig. 2.

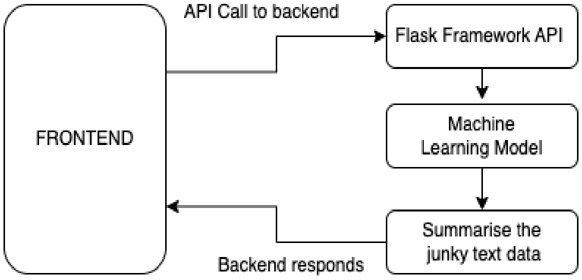


Fig. 2: Backend design

IV.DATA FLOW

The main agenda here is to summarize the junky text data. This can be achieved by creating a remote server with the capability of rephrasing the text data and the User interface using which the user should be able to send the lengthy texts and receive the rephrased summarized text using machine learning mechanism.

The flow of data is as follows:

1. User puts the data to be summarized on the Web UI developed using React Components.
2. An API Call is made to the remote server with the input text.
3. The server loads the machine learning model and them using its experience it tries to pick the unwanted text and removes the same to shorten the text data.
4. Once the rephrasing is done the processed data is sent back the user interface.

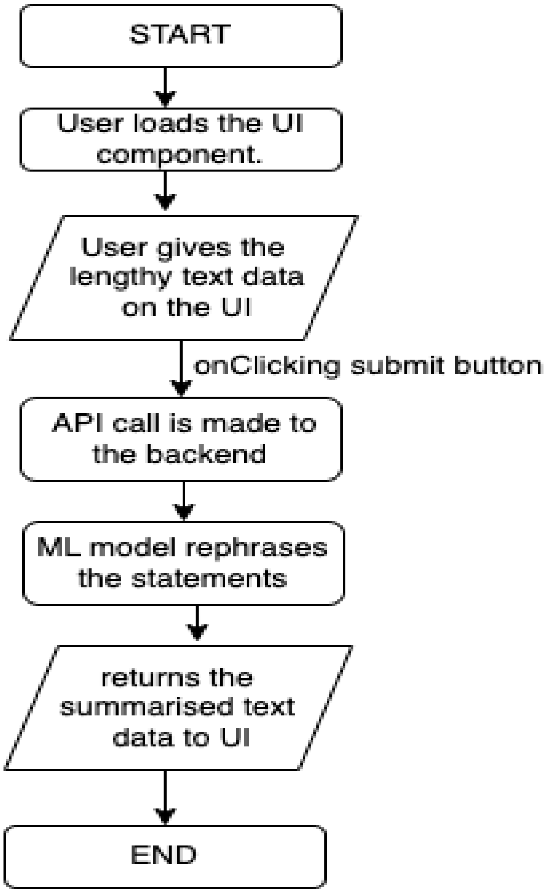

START
User loads the UI component.
User gives the lengthy text data on the UI
onClicking submit button
API call is made to the backend
ML model rephrases the statements
returns the summarised text data to UI
END

Fig 3 Data Flow Chart

V.USER INTERFACE DESIGN

The User interface majorly consists of the Body space where the user puts his original text which is supposed to be summarized. This body area will take care of the user experience by providing him/her with the various UX features like auto-complete, easy to view etc.

The navbar is designed in such a way that the title of the portal is fixed at the top orientation and the config should be accessible to the user easily. The submit button at the middle of the screen is used for the execution purpose. An API call is made to get the rephrased text as output after clicking on the submit button as shown in Fig 4. The responsiveness of the application is also important so that the user can access the application using various devices of different screen resolution. The Overall Web UI design is shown in fig 4. and Mobile UI design is shown in fig 5. This application should easily handle various screen sizes and resolution through its responsive design.

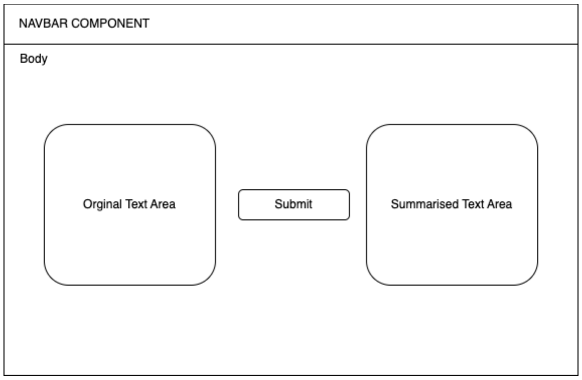


Fig 4: Web UI Design

The responsiveness of the application is also important so that the user can access the application using various devices of different screen resolution.

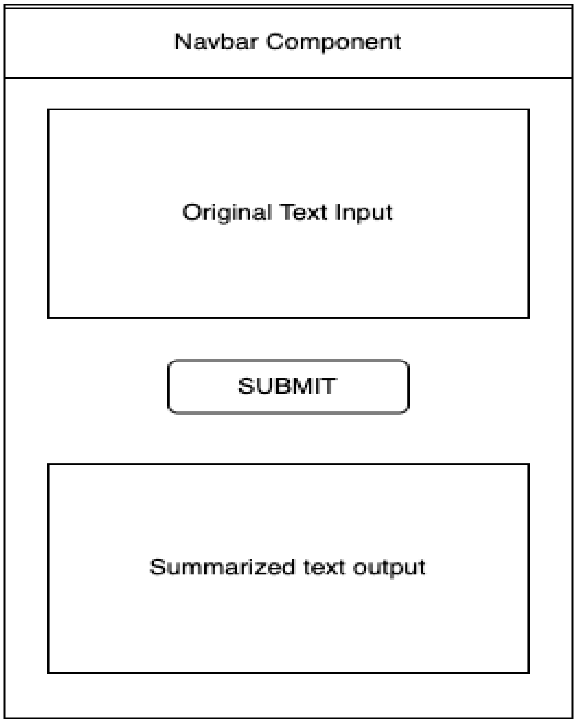


Fig 5: Mobile UI Design

VI.TESTING

The application needs to be tested at multiple level of development and post development. Unit level of testing should make sure that the each React component is rendered properly on successive flow of events like onClick, onView, onScroll etc. It's the developer's responsibility to check each component with check the props sent and the state initialized at the unit level. There will be three major divisions for the testing where one being the user interface testing, backend testing and integration testing.

A. Unit and Integration Testing

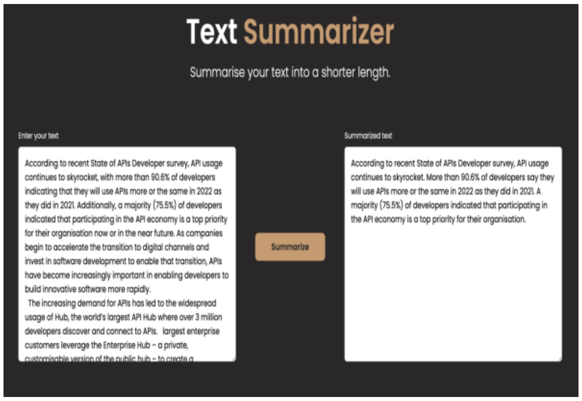
Test scenarios should be established before the execution of the test cases. Some of the testing scenarios for UI testing are as follows.

1. Navbar Component: The Navbar component should get rendered successfully.
2. OrginalText Text-area: The text-area should allow user to put his input. The input data should be stored in a state variable.
3. Submit Button: The submit button should get rendered successfully. The button should trigger the methods which would hit the API of the backend to fetch the summarized text data

VII.INTERPRETATION OF RESULTS

As reported in the earlier chapters, The design and implementation of the proposed application. The process from requirement analysis to the testing part, we conclude here to the result part. The output can be discussed under the test environment along with sample outputs with the help of screenshots captured.

The application developed to create the coding environment on web flow is as shown below fig 6



According to recent State of APIs Developer survey, API usage continues to skyrocket, with more than 90.6% of developers indicating that they will use APIs more or the same in 2022 as they did in 2021. Additionally, a majority (75.5%) of developers indicated that participating in the API economy is a top priority for their organisation now or in the near future. As companies begin to accelerate the transition to digital channels and invest in software development to enable that transition, APIs have become increasingly important in enabling developers to build innovative software more rapidly.

The increasing demand for APIs has led to the widespread usage of Hub, the world's largest API Hub where over 3 million developers discover and connect to APIs. largest enterprise customers leverage the Enterprise Hub – a private, customisable version of the public hub – to create a

According to recent State of APIs Developer survey, API usage continues to skyrocket. More than 90.6% of developers say they will use APIs more or the same in 2022 as they did in 2021. A majority (75.5%) of developers indicated that participating in the API economy is a top priority for their organisation.

Fig 6. Summarizer Text View

VIII.CONCLUSION

As An administration-organized methodology can be useful for extracting important information from text documents due to the vast amount of text information that is available on the Web but cannot be examined by people.

In general, summarization is useful because it compiles information for easier use and analysis. In this work, a text summarizer is created that can produce a summary of a Wikipedia article in the number of lines the user specifies. The proposed method is essentially an extraction-based one. The extractive approach involves selecting the top N sentences that convey the content's entire message in the best way.

Given that it is less complex than the abstractive methodology, this is the most well-known methodology. The system being proposed only works with text documents, but it is intended to eventually create text summarization models for all document types, including graphs, images, pictures, and videos in addition to text documents. Instead of using an online tool, performance evaluation of the algorithm will be conducted with human summary.

B. Future enhancements

The following are the some of the enhancements that can be made which can bring some more features to the application

- Currently it supports 500 words to shorten text. Unlimited words of input Can be implemented. Currently multiple sessions can be created in one go without saving any data. Save code feature can be implemented.
- It’s implemented for a single theme; multiple themes can be implemented.
- It has been implemented for standard mode. Formal, simple, creative and different modes can be added.
- Add view statistic option to it.

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any work would be incomplete without mentioning the people who have made it possible, I would like to thank Dr. Dhananjay Maktedar, for giving us the guidance and encouragement to complete our project successfully. It gives us immense pleasure in expressing our heartfelt gratitude for boosting our confidence and standing with us in our tough times.

REFERENCE

[1] "A Survey of Deep Learning Methods for Text Summarization" by Liu, Y., Potash, P., Lapata, M. (arXiv, 2018)

[2] "Neural Text Summarization: A Critical Evaluation" by Narayan, K., Kocisky, T., Bradbury, J., Grefenstette, E. (arXiv, 2018)

[3] "Effective Summarizing Skills" by Carolyn V. Hamilton (ThoughtCo, 2019)

[4] "Summarizing: A Guide for Readers and Writers" by Diana Hacker and Nancy Sommers (Bedford/St. Martin's, 2019)

[5] "The Art of Summarization" by Martin K. Stopfer (Scientific Writing, 2019)

[6] "Summarizing: A Guide for Readers and Writers" by Diana Hacker and Nancy Sommers (Bedford/St. Martin's, 2019)

[7] "Summarization with Pointer-Generator Networks" by See, A., Liu, P. J., Manning, C. D. (arXiv, 2017)

[8] "How to Summarize a Text" by Linda Aragoni (The Writers' Co-op, 2015)

[9] "Abstractive Text Summarization using Sequence-to-Sequence RNNs and Beyond" by Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., Zhou, B. (arXiv, 2016)

[10] "Text Summarization with Pretrained Encoders" by Zhang, Y., Gan, Z., Conneau, A. (arXiv, 2019)

[11] Browser based Code Editor By Terna Engineering College, Mumbai

[12] Portable Code Compiler published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 |Issue-4, June 2020, pp.135-138

[13] A Study and Analysis of Compiler Optimization By Authors: Prathibha A. Ballala, Dr. Sarojadevi H.

[14] Compiler and Runtime Support for Programming in Adaptive Parallel Environments Published as: University of Maryland Technical Report CSTR-3510 and UMIACS-TR-95-83

[15] Banking Interprocedural Compilation of Irregular Applications for Distributed Memory Machines Published in: Supercomputing '95, Department of

Computer Science Technical Reports CR-TR-3447

[16] Interprocedurally Partial Redundancy Elimination and Its Application to Distributed Memory Compilation Published in: Conference on Programming Language Design and Implementation '95