Brief.ly - Summarization of Web Articles and Audio Output Generation

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

S Parameswaran	**120A1089**
S Ramprakash	**120A1090**
Raunak Gurud	**120A1085**
Saurabh Shinde	**120A1104**

Name of the Mentor

Prof. Poonam Jadhav

Department of Computer Engineering

SIES GRADUATE SCHOOL OF TECHNOLOGY NERUL,

NAVI MUMBAI – 400706

ACADEMIC YEAR 2022–2023

CERTIFICATE

This is to certify that the Mini Project entitled "**Brief.ly**" is a bonafide work of the following students, submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "Bachelor of Engineering" in "Computer Engineering".

S Parameswaran	120A1089
S Ramprakash	120A1090
Raunak Gurud	120A1085
Saurabh Shinde	120A1104

(Prof. Poonam Jadhav)

Mentor

(Prof. Dr. Aparna B)

Head of Department

(Prof. Dr. Atul K)

Principal

Mini Project Approval

This Mini Project entitled "**Brief.ly**" by following students is approved for the degree of Bachelor of Engineering in Computer Engineering.

S Parameswaran	120A1089
S Ramprakash	120A1090
Raunak Gurud	120A1085
Saurabh Shinde	120A1104

Examiners

1……

(Internal Examiner Name & Sign)

2……

(External Examiner name & Sign)

Date:

Place:

Contents

ABSTRACT

Brief.ly is a Web Application that aims to retrieve a certain article, news or any textual information from the provided URL of the website and convert the same to a perfect short version summary which would be finally turned into an audio format that could be easily listened to. We intend to implement web-scraping, extractive summarization and text-to-speech conversion to give the desired output. Our main motive with this project is to spend mundane time wisely by listening to something productive for instance while traveling. Why make this project?
The major reasons to start this project are follow:

- You need a lot more than just text analysis skills when dealing with this kind of analysis task so if you're not already skilled at this kind of mechanical problem solving then these tools might not be right for you!

- Some companies charge hundreds per month/year depending on how many users are accessing their service which means that if someone doesn't use yours then there isn't much value added by paying them anyway so why bother?

- If your audio is just a few seconds long then you might need to generate more than just one summary. How about having a different length for each one? This tool can do that!

ACKNOWLEDGEMENT

We would like to express our thanks to the people who have helped us the most throughout our project. We are grateful to our guide (Prof. Poonam Jadhav) and the coordinator (Prof. Sunil Punjabi) for nonstop support for the project.

A special thanks goes to each other who worked together as a team in completing the project, where we all exchanged our own interesting ideas, thoughts and made it possible to complete our project with all accurate information. We also wish to thank our parents for their personal support and attention who inspired me to go my own way.

We would also like to extend our sincere gratitude to our Principal (Dr. Atul Kemkar) and our Head of the Department (Dr. Aparna Bannore) for their continuous support and encouragement. We would also like to thank our other faculty members for providing us with all the required resources and references for the project.

LIST OF ABBREVIATIONS

1. gTTS - Google Text-to-Speech Library
2. VSCode - Visual Studio Code
3. HTML5 - Hypertext Markup Language 5 Format
4. BS4 - Beautiful Soup 4 Library
5. WSD - Word Sense Disambiguation
6. API - Application Programming Interface
7. URL - Uniform Resource Locator
8. UI/UX - User Interface & User Experience

1.1 Introduction

Automatic extraction of key information from any source and presenting it in a concise manner can save a lot of time and resources by reducing the requirement for human intervention. Generally, summarization can be done in two ways: extractive and abstractive. In extractive summarization, words that are deemed important are extracted from the original text, while abstractive summarization summarizes the most important themes and generates new phrases and sentences. Brief.ly extracts important sentences from a given article, reports them in an order of importance (highest to lowest), and converts the final output into audio using Google Text to Speech (GTTS) library.

Summarization is the process of summarizing a document, speech or video into an efficient summary. The goal is to reduce the length of the original material without losing its essential information.

Briefly is a web application that processes HTML5 format and generates summaries in mp3 format. It can also be used on other types of media such as images and videos but we will focus on audio only here because it's much easier to understand what's going on when you only have one type of data outputting from your computer instead of trying to parse images out of YouTube videos while they're playing back in real time!

1.2 Motivation

You can use it to convert wasted time into productive time. For example, let's consider that you are traveling and you have some spare time in your hands. The problem is that there are many interesting things happening around us but we cannot see them at a glance; this is called "wasted" or "inertia".

The goal of summarization is to extract important information from a text and present it in a concise manner, so that the user does not have to read all sentences in order for him/her to understand what has happened on his/her way. For example, if someone writes "I went shopping", then we need to only look at those words: shopping...

This problem was first discussed in detail by German linguist Georg Meyers in his book "Das Problem der Zusammenfassung" (The Problem of Summaries). He wrote:

“The first point is that we are always interested in the events that happened to us during our trip, but not necessarily all the details. We have a lot of spare time and want to know what has happened on our way home. But if we receive hundreds of sentences about what people did yesterday, then it will take too much time for us to understand everything.”

This is where we decided to develop a web application where the problem is met with a solution which will combine the necessary features for summarization and audio output.

1.3 Problem Statement and Objectives

One copes up with the mundane tasks which cannot be skipped yet the time is not being used in productive manner are a big problem in the society. You need a lot more than just text analysis skills when dealing with this kind of analysis task.

In today's world, Information is the new Gold and Time is the new currency.

An application is needed to convert this unproductive time into productive time with only useful information and audio output as it should be convenient to absorb the said information.

Objectives of the proposed solutions are:

- To convert unproductive time to productive time.
- To provide relevant information to the user.
- To facilitate user comfortability by minimizing user efforts
- To develop a web extension for easy access of summarized audio output for the users.

2. Literature Survey

The Report critically reviews and summarizes the advantages of the existing systems and improvements in the following section.

2.1 Survey of Existing System/SRS

There are many research works in the area of summarization and text to speech generation. Some of them are

Extractive and Abstractive Summarization: Extractive Summarization is the process of extracting a meaningful textual representation from a large corpus in order to reduce its size; this can be done by using statistical techniques such as clustering or word sense disambiguation (WSD). In contrast, Abstractive Summarization involves generating multiple possible paraphrases for each sentence at hand, which can then be compared with user input and selected based on their similarity .

Web Article Summary: A simple approach to generating summaries from web pages has been proposed by Google Inc., where they create an automatic tool called “Google News” which summarizes news articles written about companies listed on the stock market . This method works well when there is not too much text on a page since it doesn't require any manual work by humans but would not work if you have huge amounts of data because then there would probably be too many sentences that need processing before being added up into one summary sentence."

The most common approach to summarization is phrase-based, which breaks down the text into individual sentences and then uses a method like LSA or TFIDF to select important phrases that best describe the document’s topic. This technique works well for large amounts of data but does not produce results as good as human-written summaries.

Google’s approach to summarization works by identifying important sentences in the input document and then generating a summary based on those selected sentences. The method is simple but effective, as it does not require any manual work by humans.

2.2 Limitation of Existing system or Research gap

There isn't a single system which combines all three together.

1.Web Scraping

2.Summarization

3.GTTS

This non-existence of a combined system limits an application to fulfill the needs of users who want to spend their mundane time i.e. the time used in traveling productively.

They can't read while traveling so it is not wise to just give them written articles. Making it easy to absorb is vital for the system to succeed.

There also isn't a single extension in the market. Browser extension is the perfect mesh for implementing all three together and to provide the user with all the requirements they need.

3. Proposed System

3.1 Introduction

Brief.ly is a browser extension that extracts sentences from a given article, reports them in an order of importance (highest to lowest), and converts the final output into audio using Google Text to Speech (GTTS) library. The motto of Briefly is "Spend your time wisely with Brief.ly". This can be achieved by following simple rules:

1. Extract important sentences from web pages;

2. Group similar sentences together;

3. Highlight important words within each group;
   * Report them as one sentence with all possible words listed separately.
   * Convert text into audio files using GTTS API.
   * Add background music and voiceover to the output file.

The main goal is to reduce the time spent on reading, while retaining all the information provided by the original article. The steps above are performed in sequence and can be stopped at any moment. You can modify them as you wish.

Brief.ly can be used in many ways:

- As a personal assistant by extracting important sentences from the articles you want to read and reporting them back to you;
- To provide an overview of what is happening around the world, with news from different sources displayed on one page;
- For students who have too much homework, allowing them to get a general idea about all the information they need to know for their exams;
- For businesses who want to keep track of the latest trends in their industry;
- As an online tool for making notes about your personal life, where you can write down your thoughts or quotes from books.

3.2 Architecture/ Framework

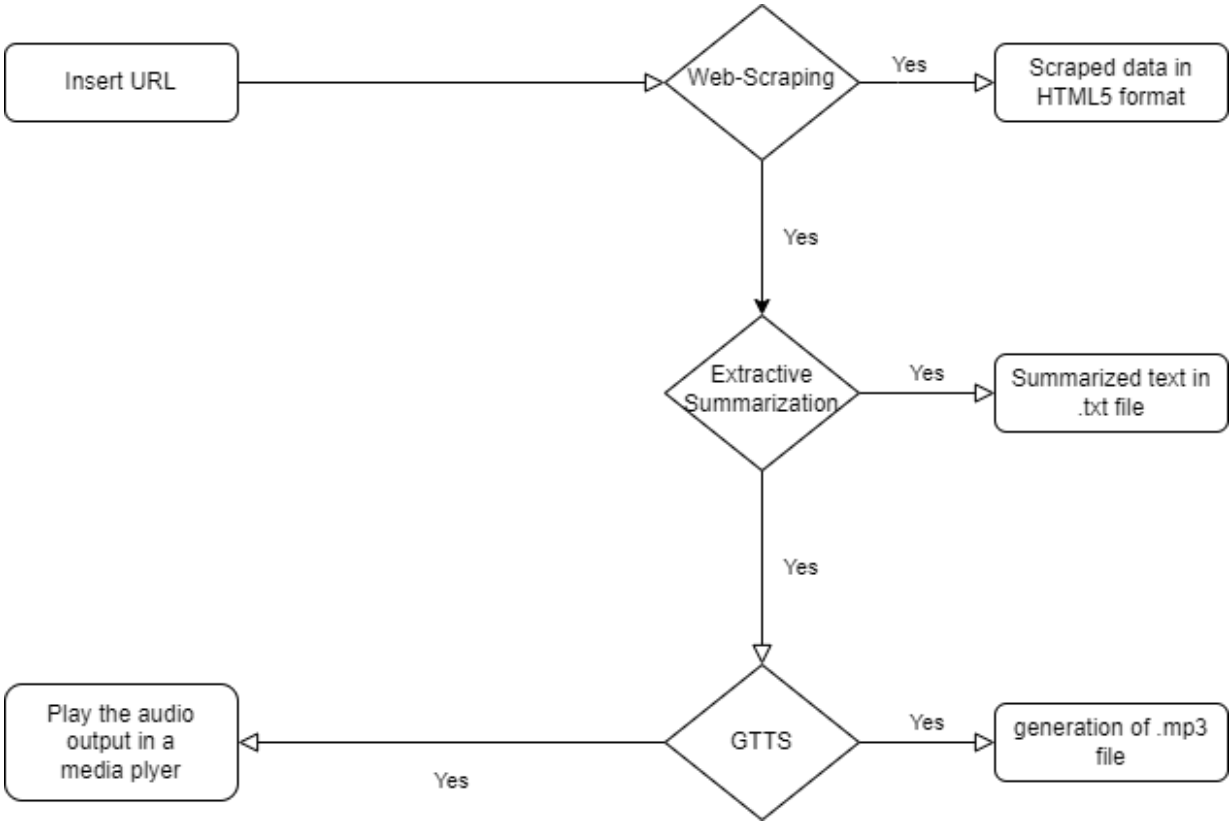


Fig2. Flowchart of Brief.ly

Flow of Brief.ly in Our Proposed System

- Go to the selected URL of the article which you want to get summarized.
- The selected article would be web scraped and the content would be in HTML5 format.
- This data will be stored in groups and it will be tokenized.
- With the help of Extractive summarization, the summarized text would be stored in .txt file

- This .txt file would be converted in audio/mp3 format using GTTS library.
- For ease access, we have build a web extension which could be opened whenever you want on your desired website
- The extension would send requests to the server.
- The server will reply to the web-client with an audio and text file.

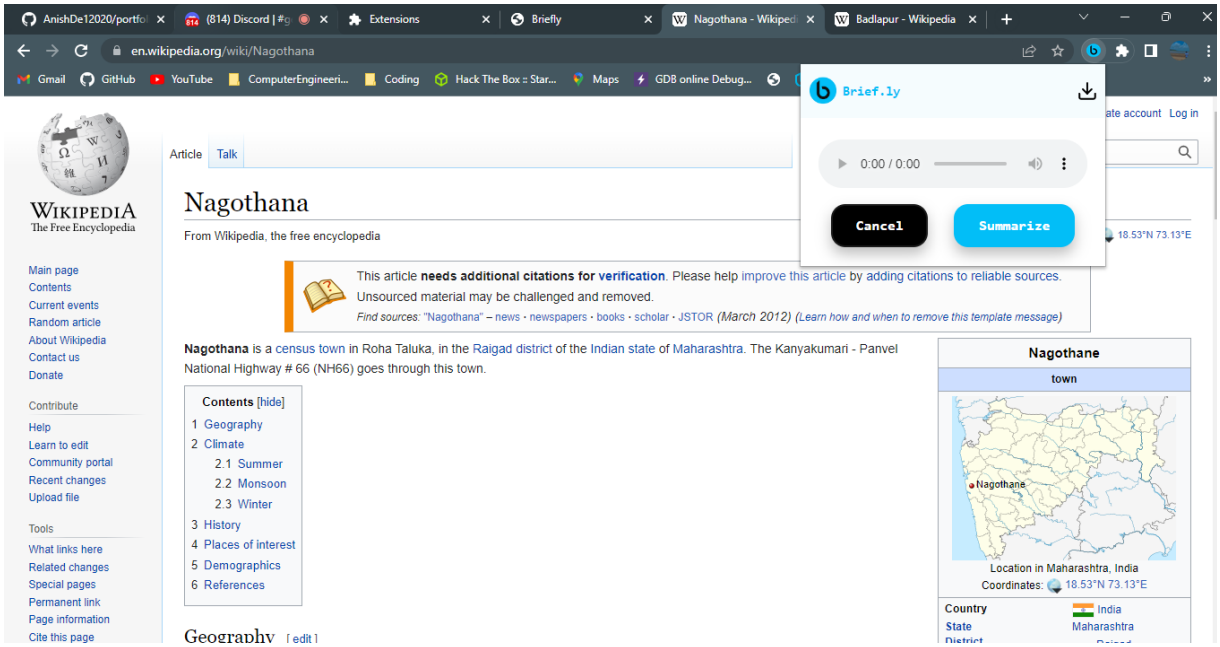
3.3 Algorithm and Process Design

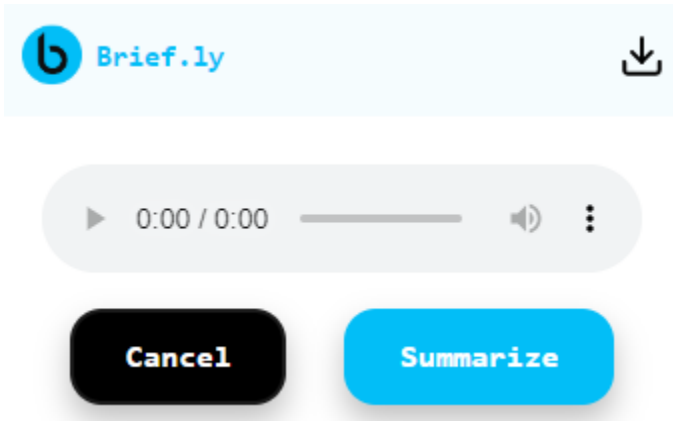
- We would First group textual data together from the article. This is called Web-content mining or web-scraping.
- Individually words are separated and tokenized.
- Each word's frequency is calculated. The frequency of the word which is highest is chosen and every word frequency is divided with that frequency.
- The values generated are between 0 and 1.
- The values closer to 1 are selected and sentences which contain them are prioritized.
- The sentences with most priority are selected and a summary is successfully extracted

3.4 Details of Hardware & Software

- gTTS
- Transformers
- BeautifulSoup4
- PyTorch
- VSCode

3.5 Experiment and Results





3.6 Conclusion and Future work

This project was created in order to explore the summarization techniques and web scraping. We were learning about web content mining in the DWM subject which made us curious. As we started creating this project, we felt the need to give it a professional background rather than a flashy user interface. Thus we created a web-application which delivered our intended objectives.

The future scopes of our project are:

1. Randomizing Articles.
2. Making it accessible for blind people through voice assistance.
3. Summarizing Video transcripts by collaborating with YouTube.
4. Improving UI based on research and user feedback.

4. References

1. A. Karmel, Anushkar Sharma, Muktak Pandya, Diksha Garg. Iot based Assistive Device for Blind People. *British Journal of Blindness Innovation & Research* [2011].

2. Chutmei Zheng, Guomei He, Zuojio Peng. A Study of Web Information Extraction Technology & Beautiful Soup. *Medium Chinese Web Research* [2008]

3. Cibambo Steven, Lew Tu, Patrick Vonplaten. Web Scraping Wikipedia Articles Using Python. *International For Research In Programming* [2012]

4. Thomas Wolf, Victor Sang, Julien Chaumond. Hugging Face Transformers and Natural Language Processing. *The Journal of Machine Learning Research* [2000]

5. Shivangi Nagdewani, Ashika Jain, Akihsa Jain . A Review on Methods for Speech-to-Text and Text-to-Speech Conversion. *International Journal of Speech Technology* [1989]