MKSSS's Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

(2022-2023)

DEPARTMENT OF

Information Technology

MINI-PROJECT REPORT

ON

'TEXT SUMMARIZATION'

SUBMITTED BY

3663_Siddhi Sonawane

3667_Vaishnvi Gilbile

3677_Nikita Palvi

UNDER THE GUIDANCE OF

Dr. Dipti Patil

ABSTRACT

Text Summarization as a phenomenon has always been present and rather an evolving one with the advent of new technologies both in terms of data collection as well for the processing of this data. One reason for using text summarization is the huge amount of data floating over the internet in the form of text files, and comments which are thought potent enough to be used to extract useful information. but since the amount of text present in these sources is too huge, so the need for text summarization becomes justified by every argument. This summarizer application takes the URL of selected data, performs summarization on the selected elements, and then presents this summarized text content on the front end of a web application. At the backend, the process of scraping web page content (if an HTTP URL is provided as input) using a beautiful soup library or reading of the text provided takes place. news in short forms, or microblogging websites.

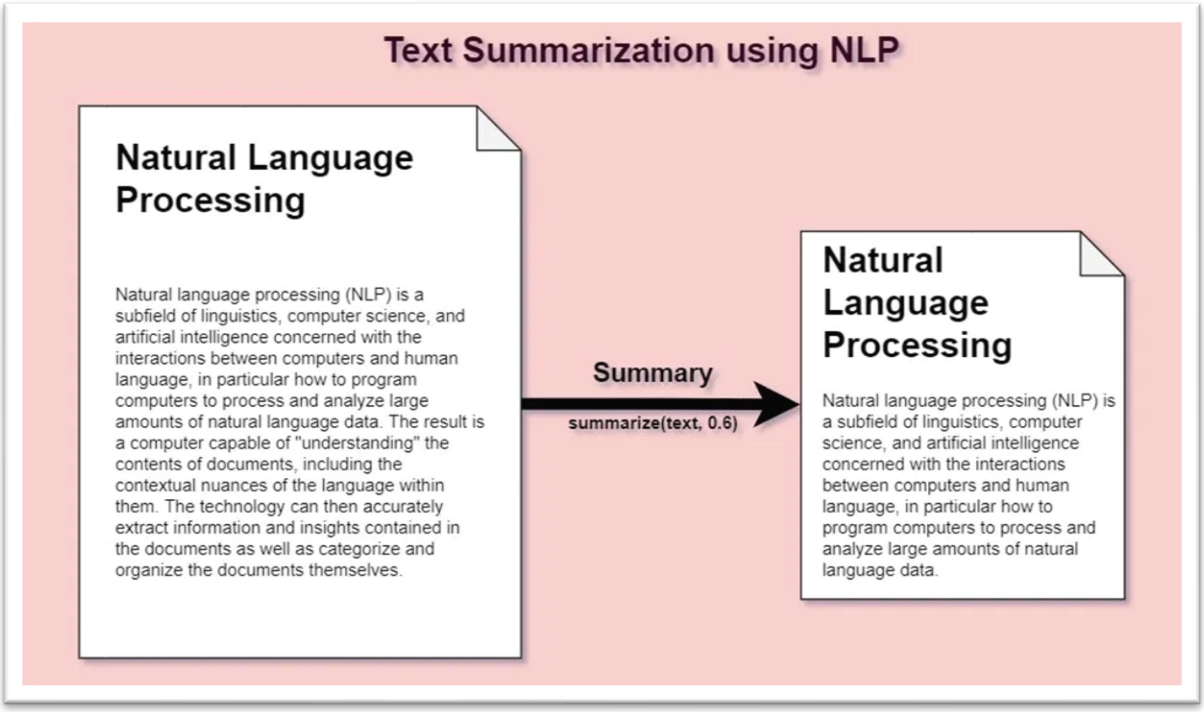
Problem Statement

The problem statement consists of text summarization for an URL using NLP. The goal of this project is to develop a natural language processing (NLP) based text summarization system that can generate a concise and informative summary of a given text document. The system should be able to take in any URL as input and output a summary that captures the most important information in the original text. The summary should be coherent, readable, and grammatically correct, and should be significantly shorter than the original text while retaining the main points.

Introduction

Text summarization in NLP means telling a long story in short with a limited number of words and conveying an important message in brief. There can be many strategies to make the large message short and give the most important information forward, one of them is calculating word frequencies and then normalizing the word frequencies by dividing by the maximum frequency.
After that find the sentences with high frequencies and take the most important sentences to convey the message.

Time optimization, as it takes lesser time to get the gist of the text in summary. Indexing can be improved with automatic summarization. More documents can be processed if we use automatic summarization. Bias is lesser in automatic summarization rather than in manual ones.


Text Summarization using NLP
Natural Language Processing
Summary
summarize(text, 0.6)
Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

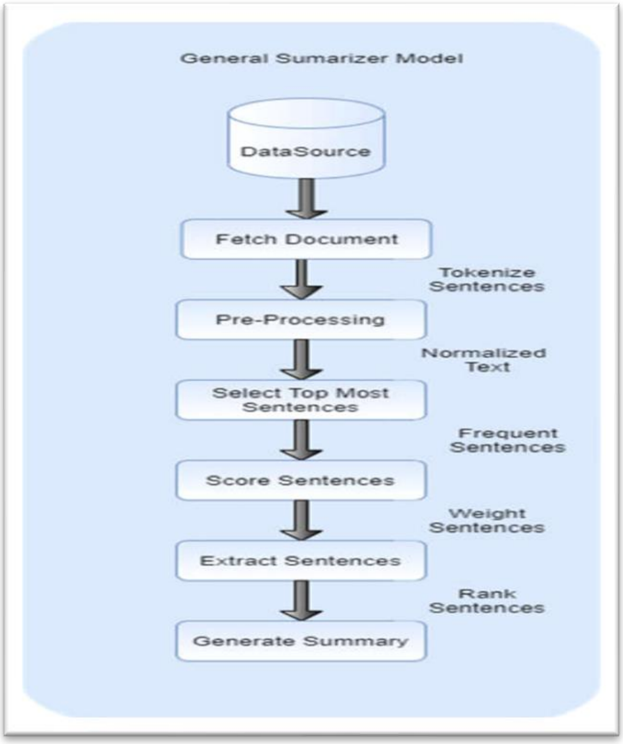
Objective

NLP text summarization is the process of breaking down lengthy text into digestible paragraphs or sentences. This method extracts vital information while also preserving the meaning of the text. This reduces the time required for grasping lengthy pieces such as articles without losing vital information. Abstractive text summarization is a complex task whose goal is to generate a concise version of a text without necessarily reusing the sentences from the original source, but still preserving the meaning and the key contents.

Requirement ;

- Python version > = 3.0
- Spacy v3.10
- Flask
- Beautiful Soup v4.0 or above

Flowchart of Text Summarization :



Working Methodology

Steps to Text Summarization:

1) Text Cleaning: Removing stop words, and punctuation marks and making the words in lower case.
2) Work Tokenization: Tokenize each word from sentences.
3) Word Frequency table: Count the frequency of each word and then divide the maximum frequency with each frequency to get the normalized word frequency count.
4) Sentence Tokenization: As per the frequency of the sentence then
5) Summarization

How do I run it :

1. Run app.py
2. Click on the link displayed in the terminal
3. Paste the URL of the article in the URL filed.
4. Click the summarize button to get a summary.

How does it work?

Upon receiving a URL, scraper.py scrapes the text present on the website. The text is then formatted and the client.

This formatted text is then passed to the summarizer.py which uses spacy to tokenize the text into sentences and words.

The frequency of each word is calculated and stored in a dictionary. The frequency of each word is then normalized by dividing by the maximum frequency (this is done in order to find the relative frequency of each word).

Next, the sentence scores are calculated by adding the word frequency of each word present in the sentence.

A heap queue is then used to get sentences with the highest sentence scores.

The sentences are then joined to get the summary.

Next, the estimated reading time is calculated.

Finally, the title, summary, and estimated reading time are displayed.

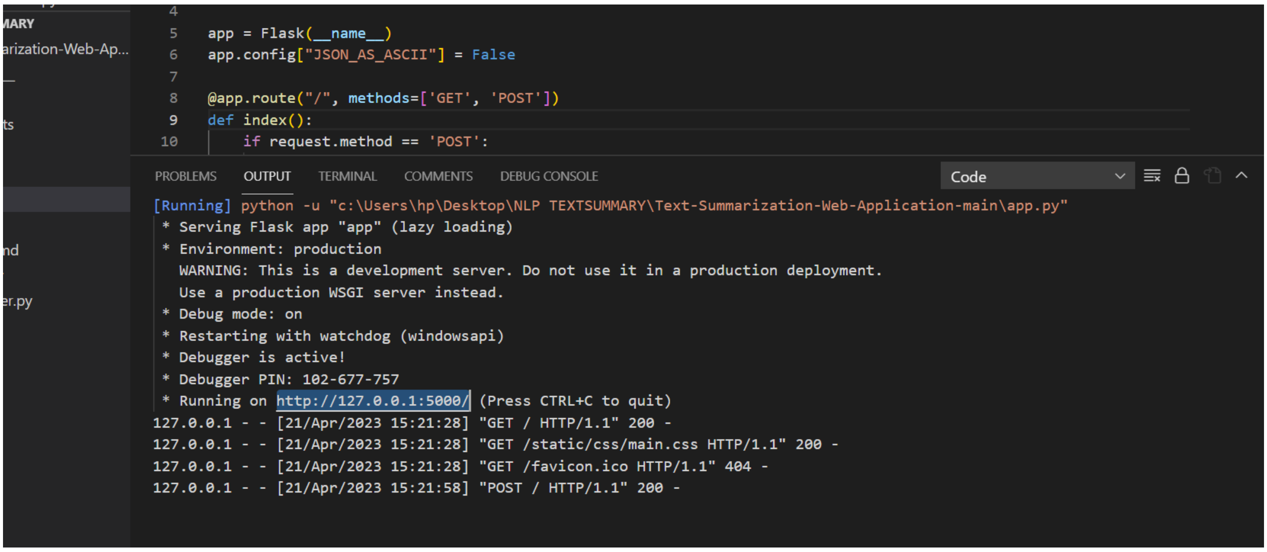
Imported libraries

```
from flask import Flask, render_template, request, url_for
from scraper import scraper
from summarizer import summarizer, estimated_reading_time
```

```
import bs4 as bs
import urllib.request
import re
```

```
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
# Used to rank sentences according sentence scores
from heapq import nlargest
```

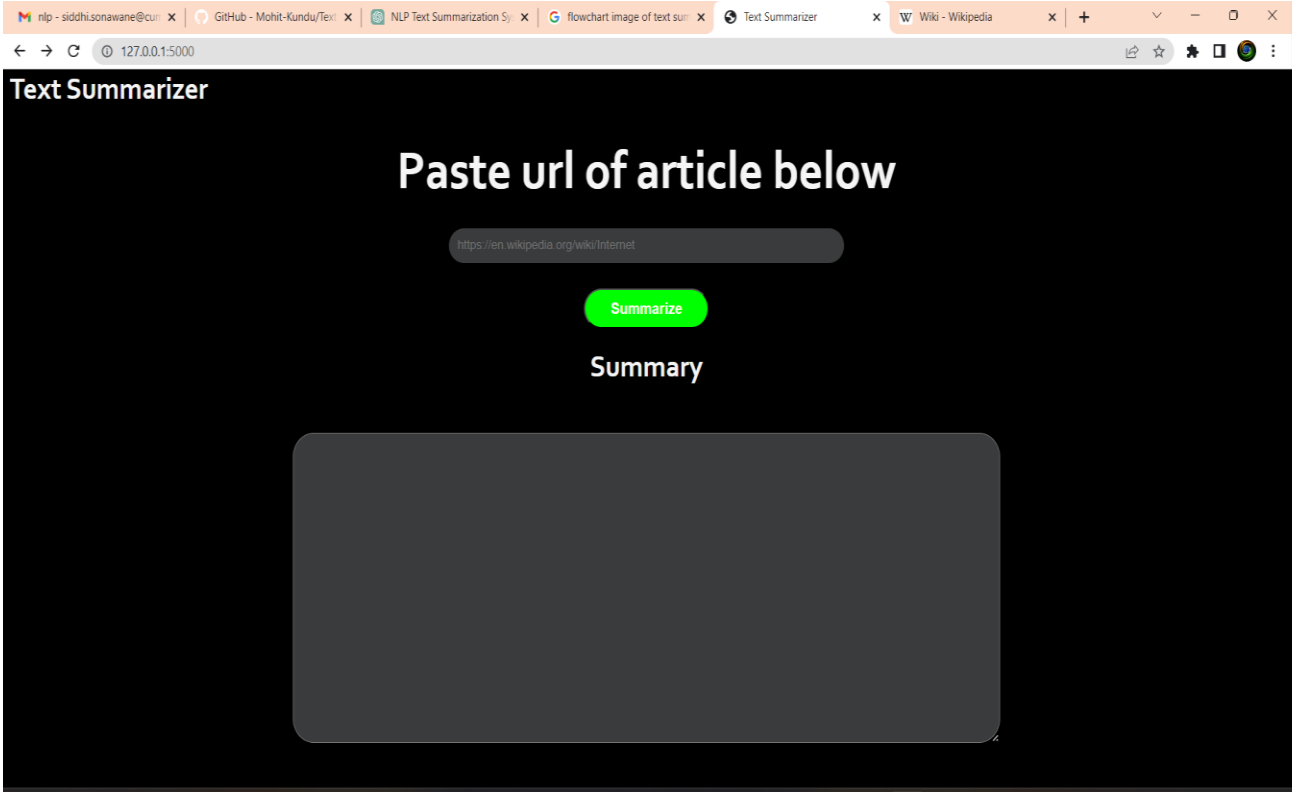
Output :

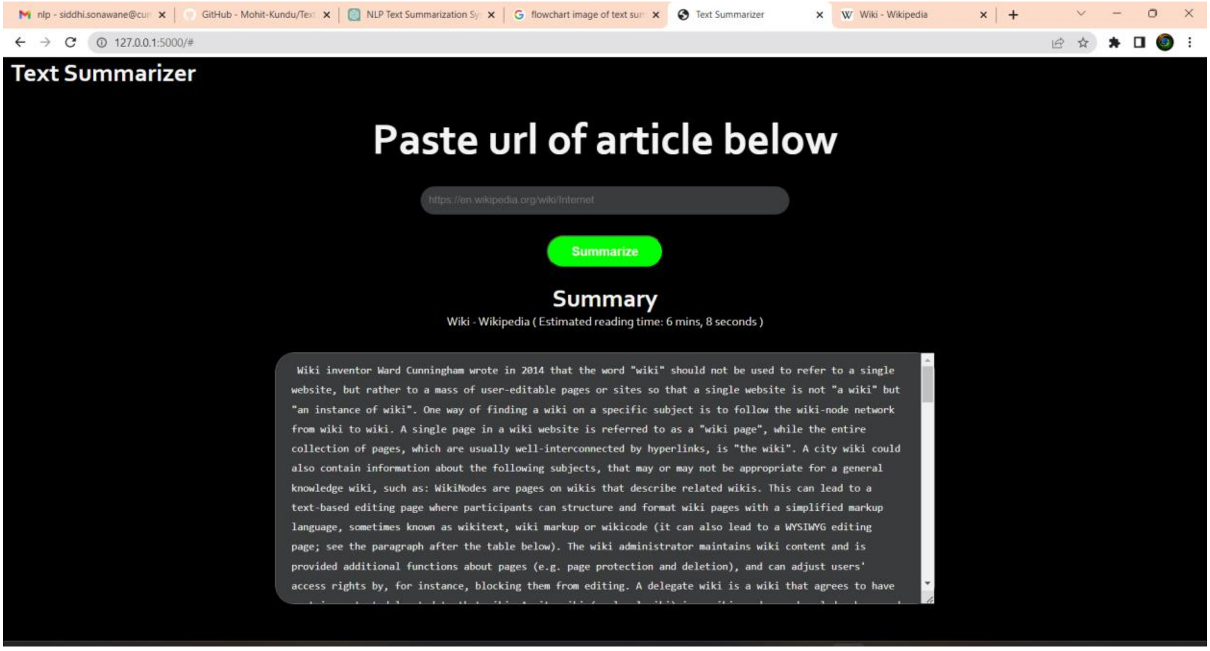


```
app = Flask(__name__)
app.config["JSON_AS_ASCII"] = False

@app.route("/", methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
```

```
[Running] python -u "c:\Users\hp\Desktop\NLP TEXTSUMMARY\Text-Summarization-Web-Application-main\app.py"
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 102-677-757
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Apr/2023 15:21:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Apr/2023 15:21:28] "GET /static/css/main.css HTTP/1.1" 200 -
127.0.0.1 - - [21/Apr/2023 15:21:28] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [21/Apr/2023 15:21:58] "POST / HTTP/1.1" 200 -
```




nlp - siddhi.sonawane@cur
GitHub - Mohit-Kundu/Text
NLP Text Summarization Sy
flowchart image of text sur
Text Summarizer
Wiki - Wikipedia
127.0.0.1:5000/#
Text Summarizer
https://en.wikipedia.org/wiki/Internet
Summarize

Paste url of article below

Summary

Wiki - Wikipedia (Estimated reading time: 6 mins, 8 seconds)

Wiki inventor Ward Cunningham wrote in 2014 that the word "wiki" should not be used to refer to a single website, but rather to a mass of user-editable pages or sites so that a single website is not "a wiki" but "an instance of wiki". One way of finding a wiki on a specific subject is to follow the wiki-node network from wiki to wiki. A single page in a wiki website is referred to as a "wiki page", while the entire collection of pages, which are usually well-interconnected by hyperlinks, is "the wiki". A city wiki could also contain information about the following subjects, that may or may not be appropriate for a general knowledge wiki, such as: WikiNodes are pages on wikis that describe related wikis. This can lead to a text-based editing page where participants can structure and format wiki pages with a simplified markup language, sometimes known as wikitext, wiki markup or wikicode (it can also lead to a WYSIWYG editing page; see the paragraph after the table below). The wiki administrator maintains wiki content and is provided additional functions about pages (e.g. page protection and deletion), and can adjust users' access rights by, for instance, blocking them from editing. A delegate wiki is a wiki that agrees to have

Advantages :

- Wide range of sources: There is a vast amount of information available on the internet, and summarizing web pages using their URLs allows access to a wide range of sources that can be summarized for various purposes.
- Faster summarization: Summarizing a web page using its URL can be faster than manually reading and summarizing the content, especially when dealing with large amounts of data.
- Cost-effective: Since web pages are publicly available, there is no need to pay for access to content, which makes text summarization using URLs a cost-effective approach.
- Accessible from anywhere: Summarizing web pages using their URLs can be done from any device with an internet connection, making the process more accessible.
- Automated summarization: Once a system is developed and trained, summarization using URLs can be fully automated, reducing the need for human intervention.
- Consistency: Automated summarization using URLs can produce consistent results, which can be helpful for creating reports or analysis.

Disadvantages :

- Domain-specific terminology: Domain-specific terminology can be difficult for an NLP model to accurately summarize, especially when dealing with technical or scientific terms.

- Bias: Depending on the training data and the model used, there is a potential for bias in the generated summaries, which can affect the accuracy and objectivity of the information.

- Over-summarization or under-summarization: The summarization process can result in either over-summarization, where important information is left out, or under-summarization, where the summary contains irrelevant or redundant information.

- Quality of content: Web pages can have varying levels of quality and relevance, which can affect the quality and accuracy of the generated summary.

- Lack of context: Summarizing a web page using its URL may not provide enough context for the summary, which can lead to incomplete or misleading summaries.
- Language barriers: Summarizing web pages from different languages can be challenging and may require additional processing to accurately capture the meaning.

Applications of Text Summarization

1. News: This technique has multiple applications in the field of News. It includes creating an introduction, Generating headlines, and Embedding captions on pictures.

2. Scientific Research: Algorithms are used to dig out important information from Scientific research papers. AI is outranking human beings in doing so.

3. Social Media Posting: Content on Social media is preferred to be concise. Companies use this technique to convert long blog articles into shorter ones suited for the audience.

4. Creating Study Notes: Many applications use this process to create student notes from vast syllabus and content.

5. Conversation Summary: Long conversations and meeting recording could be first converted into text and then important information could be fetched out of them.

6. Movie Plots and Reviews: The whole movie plot could be converted into bullet points through this process.

7. Deliverable Feeds: They are the short piece of information derived from the complete informative articles. These are generally delivered to people through emails or feed delivery services.

8. Content Writing: Not from the scratch though but on providing a topic and points an outlined summary could be generated.

Thank you