SummarizeMe

TEXT SUMMARIZATION USING NLP

Submitted By:
Abhishek Joshi
500090966
B. Tech CSE (Spl. AI&ML) [H]

Submitted To:
Dr. Touseef Iqbal

C O N T E N T S

PROLOGUE

In today's digital age, we're constantly bombarded with information. From online articles to research papers, the sheer volume of text can be overwhelming. Summarization tools can help us quickly grasp the key points of long pieces of text, saving us time and improving our comprehension.

PROBLEM STATEMENT

Manually summarizing lengthy text can be a tedious task. Traditional methods require careful reading, identifying important information, and condensing it into a concise format. This process can be time-consuming and error-prone.

SummarizeMe aims to address this challenge by providing an automated solution for text summarization.

APPROACH

SummarizeMe leverages the power of Natural Language Processing (NLP) to analyze text and generate summaries. In this project, I implemented a summarization technique that scores sentences based on word frequency and sentence length. Sentences with more frequent and relevant words are considered more important for the summary.

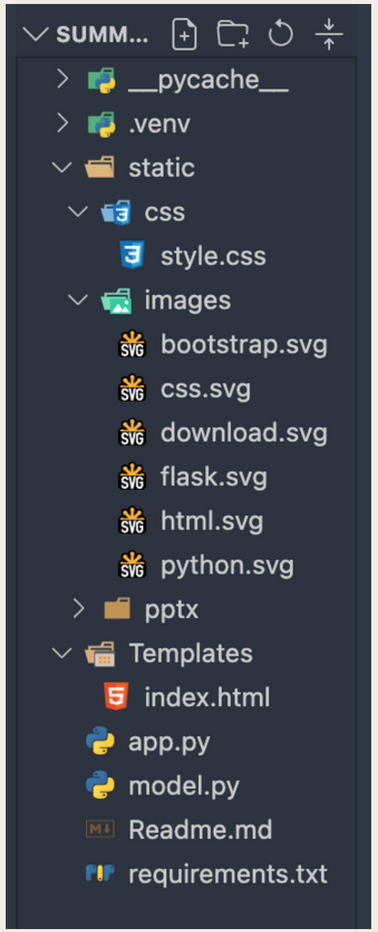
There are other summarization techniques out there, such as statistical methods that analyze word co-occurrence or graph-based approaches that consider the relationships between sentences. However, for this project, I focused on a simpler yet effective sentence scoring approach.

IMPLEMENTATION

SummarizeMe is a web application built using Flask, a popular Python framework for creating web applications. The user interface is designed with HTML, CSS, and Bootstrap for a user-friendly experience.

The application consists of two main parts:

- SummarizerModule: This Python class handles the core text processing tasks. It removes stop words (common words like "the" and "a"), tokenizes the text into sentences, calculates word frequencies, scores sentences based on our chosen criteria, and finally generates a concise summary.
- Flask App: This part defines the routes for user interaction. Users can paste their text and specify the desired number of lines for the summary. The Flask app then calls the SummarizerModule to generate the summary and displays it on the webpage.



app.py

```python
from flask import Flask, request, render_template
from model import SummarizerModule

app = Flask(__name__)
summarizer_module = SummarizerModule()

# routes
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/summarize', methods=['POST','GET'])
def summarize():
    if request.method == 'POST':
        text = request.form['text']
        lines = int(request.form['lines'])
        summarized_text = summarizer_module.generate_summary(t
        return render_template('index.html', text=text, lines=

if __name__ == '__main__':
    app.run(debug=True)
```

model.py

```python
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from heapq import nlargest
import nltk

nltk.download('stopwords')
nltk.download('punkt')

class SummarizerModule:
    def __init__(self):
        pass

    def calculate_sentence_scores(self, sentences, word_frequencies):
        sentence_scores = {}
        for sent in sentences:
            for word in word_tokenize(sent.lower()):
                if word in word_frequencies:
                    if len(sent.split(' ')) < 30:
                        if sent not in sentence_scores:
                            sentence_scores[sent] = word_frequencies[word]
                        else:
                            sentence_scores[sent] += word_frequencies[word]
        return sentence_scores

    def generate_summary(self, text, num_sentences):
        stop_words = set(stopwords.words('english'))
```

TECH STACK

SummarizeMe utilizes a strong technology stack, including Python, Flask for robust text summarization. The frontend is developed with HTML, CSS, and Bootstrap, ensuring a visually appealing and responsive user interface. This combination of technologies enables SummarizeMe to deliver accurate and user-friendly solutions for enhancing written communication.

RESULTS AND LIMITATIONS

SummarizeMe provides a convenient way to summarize text. However, it's important to acknowledge its limitations. The current version might struggle with complex sentence structures, nuanced language, or factual topics that require deeper understanding. Additionally, there's always a chance of factual errors in the generated summaries.

FUTURE OPPORTUNITIES

There's always room for improvement. Here are some exciting possibilities for future development:

- Implementing more sophisticated summarization algorithms like those based on machine learning.
- Allowing users to choose the summarization style (e.g., factual, extractive, abstractive).
- Integrating SummarizeMe with other platforms like research tools or educational software.

CONTRIBUTIONS

I've made SummarizeMe's code accessible on GitHub, inviting open source contributions. This collaborative environment fosters innovation and allows developers to enhance the tool's capabilities together. Join us in improving SummarizeMe and making it even more valuable for users.

GitHub Link: https://github.com/AbhishekJ24/SummarizeMe-NLP-Text-Summarizer.git

Thank you!