Wiki Page Summarize

Project Report

submitted by

MUVVALA N V SAI GURUDEEP (-)

Mahindra University

Bachelor of Technology
in
Artificial Intelligence

October 2023

CONTENTS

ABSTRACT

The project utilized BeautifulSoup 4, a Python library known for its web scraping capabilities. The chosen Wikipedia page, "https://en.wikipedia.org/wiki/Alexander_the_Great," was scraped, and its content was retrieved, with special attention to maintaining the hierarchical structure of headings and sub-headings.

The process can be summarized as follows:

1. Web Scraping: We employed BeautifulSoup 4 to scrape the content of the Wikipedia page. This involved extracting not only the textual content but also the headings and sub-headings, thus preserving the document's original organization.

2. Text Summarization: After successfully scraping the data, the next step was to summarize the text under each section while ensuring that the original headings and sub-headings remained intact. To achieve this, we employed prompt chains for generating summaries that maintained coherence and context.

The project's expectations included achieving proficiency in web data scraping and the ability to summarize textual information while retaining the structural integrity of headings and sections. To evaluate the quality of the summarized content, several metrics were proposed, considering factors such as:

1. Information Retention: The extent to which the key information from the original text is preserved in the summaries.

2. Readability: The clarity and ease with which the summaries can be understood by a general audience.

3. Coherence: The logical flow of information within and across sections, ensuring that the summaries are contextually accurate.

In conclusion, this report outlines the successful completion of the web scraping and text summarization project. The proposed metrics provide a basis for evaluating the quality of the summarized content, ensuring that it meets the company's expectations for information retention, readability, and coherence.

CHAPTER 1
PROBLEM DEFINITION

The problem addressed in this report revolves around the need to efficiently extract and summarize structured textual content from web sources, particularly Wikipedia pages, while preserving the original hierarchy of headings and sub-headings.

1.1 CHALLENGES

This task encompasses two primary challenges:

1. Web Scraping: Web scraping involves retrieving content from websites in a structured manner. The problem is to extract not only the main textual content but also the hierarchical structure of headings and sub-headings from a given webpage.

2. Text Summarization: Once the content is scraped, the challenge is to generate concise and coherent summaries for each section while retaining the original headings and sub-headings. These summaries should provide an accurate representation of the information within each section.

1.2 KEY COMPONENTS

The problem definition entails the following key components:

- Structural Preservation: It is essential to maintain the original structure of the document, including headings and sub-headings, in the summarization process. This preserves the logical organization of the content.

- Coherence and Context: The generated summaries should not only be concise but also coherent and contextually accurate. The challenge is to create summaries that make sense in the context of the original content and provide a clear understanding of each section.

- Metrics for Evaluation: A set of metrics needs to be defined to assess the quality of the generated summaries. These metrics should consider factors such as information retention, readability, and coherence to determine the effectiveness of the summarization process.

Addressing this problem is vital for various applications, including information retrieval, content aggregation, and knowledge extraction.

CHAPTER 2
IMPLEMENTATION

2.1 DATA SCRAPING AND CLEANING

Initially, data retrieval involved web scraping of a Wikipedia URL, ensuring the extraction of both headings and subheadings along with their corresponding paragraph text.

Following data acquisition, a cleaning process was executed to enhance data quality. This encompassed the removal of bracketed content that redirected to links, elimination of backslashes, elimination of duplicate entries, and addressing any missing values (NaN).

2.2 TEXT SUMMARIZATION APPROACHES

In the realm of text summarization, two distinct methodologies were explored:

1. **Extractive Text Summarization**:This traditional approach focuses on identifying and extracting the most salient sentences from the source text. The resultant summary comprises exact sentences directly extracted from the original content. Implementations included methods such as gensim.summarization and various algorithms available in the sumy library (e.g., LexRank, LSA, Luhn, KL).

2. **Abstractive Text Summarization**: Abstractive summarization, a more advanced approach, involves identifying key sections, comprehending context, and generating concise sentences in the summary. This method ensures that essential information is conveyed while not necessarily using verbatim text from the source. OpenAI models were employed to perform abstractive text summarization.

2.3 MODEL LIMITATION AND SOLUTIONS

A limitation encountered during the project pertained to the usage of the GPT-3.5 Turbo model for summarization due to its input token size restriction. Similar constraints were observed with other models such as Bard and T5. Two strategies were devised to overcome this limitation:

1. **Method 1**: Research and identify models capable of processing larger inputs to accommodate the data.

2. **Method 2**: Break the paragraphs into multiple strings of a manageable length for the model, summarize them individually, and subsequently combine them. If needed, perform summarization again on the concatenated strings to ensure coherence.

The chosen approach throughout this report was Method 1. After researching OpenAI models, the ***"gpt-3.5-turbo-16k-0613"*** model was identified as a solution, as it could process inputs of up to 16,385 tokens. Again choosing any 2 approached will led to a branching into different implementation approaches:

- **Summarize using Pretrained Model**: Summarization solely utilizing the capabilities of the pretrained model.
- **Summarize using Pretrained Model with a Prompt**: Summarization guided by providing a specific prompt to the model.
- **Summarize using Pretrained Model with a Chain of Prompts (ROI)**: Employing a chain of prompts to guide the model's summarization process, a technique that yielded superior results compared to the previous two methods in this report.

In this report, we adopted both Method 2 and Method 3 for our text summarization tasks. After thorough experimentation and evaluation, it became evident that Method 3 outperformed Method 2 in terms of summarization quality. To quantify the improvements and assess the efficacy of our summarization techniques, we employed the Rouge Metric (Recall-Oriented Understudy for Gisting Evaluation) as our evaluation criterion.

2.4 FIGURES

Below is our Implementation Approach. Figure 2.1

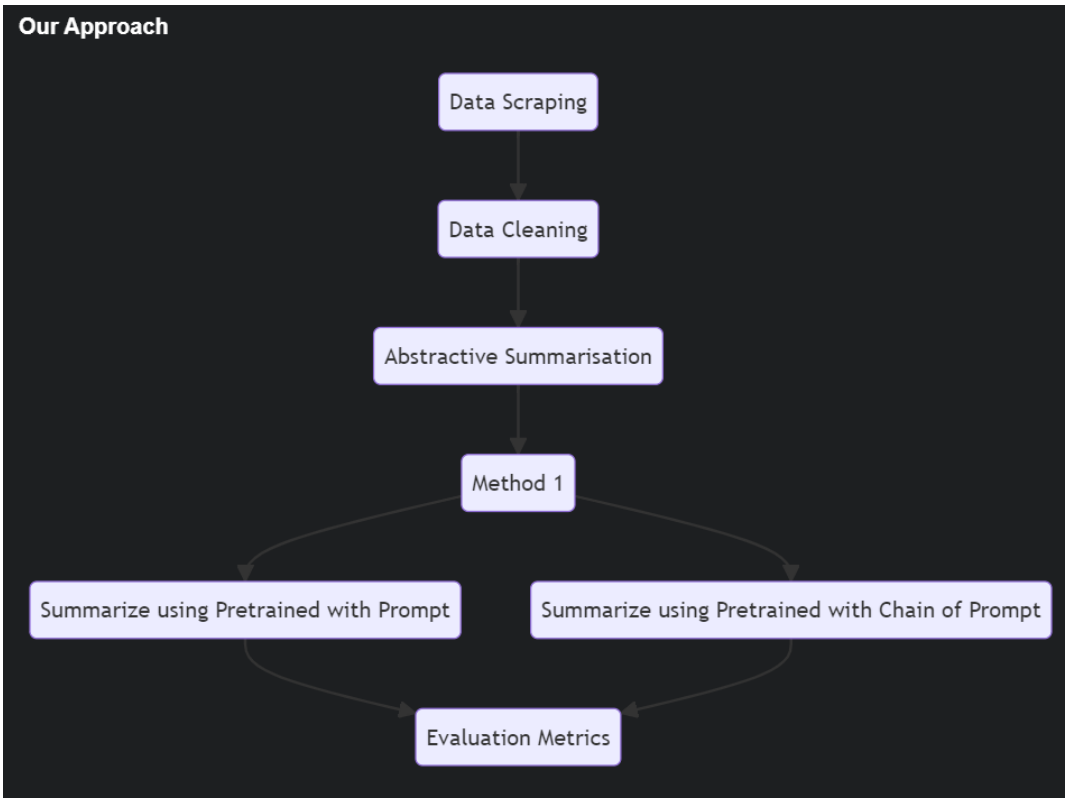

Figure 2.1: Our Implementation Approach

CHAPTER 3
PROMPT DEVELOPMENT

In the initial phase of this project, I experimented with various prompts for text summarization. Some of the prompts I explored included

Other prompts which I tried are:

```
summarize the text
summarize the text in less than 100 words
summarize the text in less than 500 words
summarize the text in less than 100 words along with retaining
maximum information
summarize the text in less than 500 words along with retaining
maximum information
```

During this experimentation, I encountered rate limit errors, which prompted me to introduce a time gap between successive prompts. It became evident that concise summaries, typically under 80 words, are more effective in quickly conveying information.

As a result, I narrowed down my approach to using the prompt ”summarize the text in less than 80 words while retaining maximum information.” To assess the quality of these summaries, I conducted a word token frequency analysis.

Subsequently, I incorporated the concept of prompt chaining into the summarization process. This approach was influenced by research on Prompt Chaining and aimed to generate coherent and contextually accurate summaries. The following prompt chain was formulated:

```
In the process of crafting progressively succinct,
information-rich summaries of the Wikipedia page dedicated to
Alexander the Great, iterate through the following steps five
times:

Step 1. Identify 1-3 pertinent Entities (separated by ';')
from the Wikipedia page that have not been included in the
previously generated summary.
Step 2. Create a new, more compact summary of the exact same
length as the previous one, encapsulating not only all entities
and details from the preceding summary but also the Missing
```

```
Entities.

Criteria for a Missing Entity:
Relevance: The entity should directly relate to the life and
accomplishments of Alexander the Great.
Specificity: It should be descriptive, coherent, contextually
accurate, and concise (limited to 5 words or fewer).
Novelty: The entity must not have been included in the previous
summary.
Fidelity: It should be present in the Wikipedia page.
Placement: The entity can be located anywhere within the
Wikipedia page.

Additional Guidelines:
-- The initial summary should be a paragraph with minimal
specific information beyond the Missing Entities. You can
employ verbose language and fillers
(e.g., 'the page discusses') to achieve the target word count.
-- Ensure that every word serves a purpose; refine the previous
summary for improved flow and to create space for new entities.
-- Utilize techniques like fusion, compression, and the removal
of redundant phrases such as 'the page discusses' to maximize
conciseness.
-- The summaries should evolve into highly concise,
self-contained, coherent, and contextually accurate narratives
that can be easily comprehended without the need to reference
the Wikipedia page.
-- The Missing Entities can be integrated at any point in the
new summary."
-- Never drop entities from the previous summary. If space
cannot be made, add fewer new entities. Remember to use the
exact same number of words for each summary. Answer in JSON.
The JSON should be a list (length 5) of dictionaries whose keys
are "Missing_Entities" and "Denser_Summary."
```

This approach proved to be successful in creating a coherent chain of prompts for generating informative and concise summaries of the Wikipedia page on Alexander the Great.

CHAPTER 4
RESULTS

4.1 ANALYSIS OF METRICS

In the evaluation of the text summarization, several widely recognized metrics can be employed to quantify the quality of the language model's output and track improvements over time. One of the metric which was choosen was **ROUGE**. This metric include:

1. **ROUGE**: ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is among the most popular metrics for assessing summarization quality. It quantifies the n-gram overlap between the generated summary and the reference summary. ROUGE is calculated for various n-gram sizes, including 1-gram (unigram), 2-gram (bigram), and 3-gram. The scores from these calculations are combined to produce an overall score.

2. **ROUGE-1 (Unigram)**: Measures the overlap of individual words (unigrams) between the generated summary and the reference summary, expressing the proportion of words in the generated summary present in the reference summary.

3. **ROUGE-2 (Bigram)**: Measures the overlap of consecutive word pairs (bigrams) between the generated summary and the reference summary, indicating the proportion of bigrams in the generated summary also found in the reference summary.

4. **ROUGE-L**: Assesses similarity by analyzing the longest sequence of words shared between the generated abstract and the reference abstract. ROUGE-L differs from ROUGE-1 and ROUGE-2 as it employs a string matching approach, not a simple word count.

5. **ROUGE-Lsum**: A variation of ROUGE-L that divides the generated and reference summaries into sentence units and gauges similarity between these sentence units.

The original ROUGE paper suggests that 'ROUGE-2 and ROUGE-L worked well in single-document summarization tasks, while ROUGE-1 and ROUGE-L perform great in evaluating short summaries.

Additionally, beyond automated metrics, manual evaluation plays a crucial role in

assessing summary quality. When manually inspecting a summary, it is essential to consider not only adherence to specific expectations such as style, format, or length but also the following general qualities:

1. **Fluency**: Ensure that the summary contains no formatting issues, capitalization errors, or ungrammatical sentences.

2. **Coherence**: Check that the summary is well-structured and organized, building logically from sentence to sentence to form a coherent narrative about the topic.

3. **Coverage**: Verify that the summary includes all the essential information from the source document or conversation.

4. **Relevance**: Ensure that the summary includes only pertinent information from the source, avoiding redundancies.

5. **Hallucinations**: Confirm that the summary does not contain incorrect information that is unsupported by the source.

4.2 MACHINE EVALUATION

To determine which ROUGE variant was most applicable, a plot of summary lengths was created, demonstrating the variation in summary length and its frequency. This analysis indicated that the summaries tended to be relatively short, suggesting that ROUGE-1 scores would be more appropriate for evaluation [Figure 4.1]

The obtained ROUGE scores for different summarization approaches are as follows:

Summarization Approach	**Avg Precision**	**Avg Recall**	**Avg F1**
Normal Prompt	0.0262	**0.8262**	0.0492
Normal Prompt with Restriction	0.0708	0.7357	0.1181
Prompt Chaining Approach	**0.0869**	0.7195	**0.1439**

Table 4.1: ROUGE-1 Scores for Different Summarization Approaches

The analysis revealed that when restricting the word length to 80 characters (summaries_normal_prompt_restrict) and employing the prompt chaining technique (summaries_chain_prompt_len), the generated summaries performed comparatively well. However, it's noteworthy that the general summaries generated using the "Summarize the Text" prompt (summaries_normal_prompt) achieved the highest ROUGE scores due to a higher number of string matches.

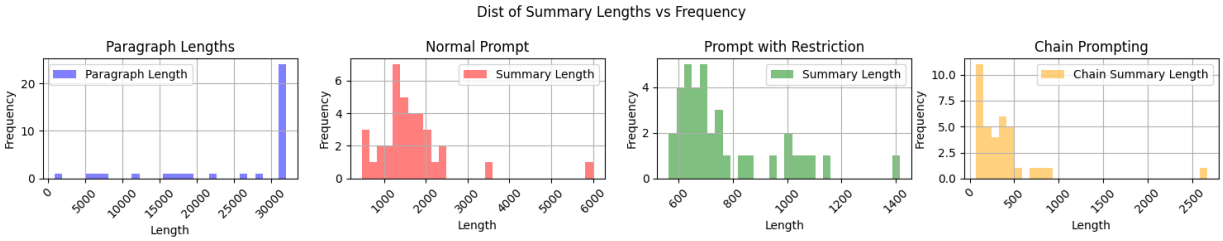

Figure 4.1: Summary length and its frequency

4.3 MANUAL EVALUATION

During manual evaluation with students, there were varied responses regarding the preference for summaries generated with the "Summarize the Text in Less than 80 Words" prompt (summaries_normal_prompt_restrict) or the prompt chaining approach (summaries_chain_prompt_len). However, the general summaries generated using the "Summarize the Text" prompt (summaries_normal_prompt) appeared to be consistently favored.

In conclusion, there is potential for further experimentation with different methods or a combination of methods to enhance the prompt chain's summarization capabilities. This iterative process aims to refine the quality of machine-generated summaries, aligning them more closely with human expectations and improving their fluency, coherence, coverage, relevance, and absence of hallucinations.

REFERENCES

In the course of completing this project, a comprehensive list of references has been consulted and cited. The following section provides an organized compilation of the sources that have contributed to the research and content development for this project:

4.4 CITATIONS

- Lin, Chin-Yew. “Rouge: A package for automatic evaluation of summaries.” Text summarization branches out. 2004.
- Adams, Griffin, et al. ”From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting.” arXiv preprint arXiv:2309.04269 (2023)..
- Liu, Yixin, et al. “On Learning to Summarize with Large Language Models as References.” arXiv preprint arXiv:2305.14239 (2023).
- Liu, Yixin, et al. “Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation.” arXiv preprint arXiv:2212.07981 (2022).
- Luo, Zheheng, Qianqian Xie, and Sophia Ananiadou. “Chatgpt as a factual inconsistency evaluator for abstractive text summarization.” arXiv preprint arXiv:2303.15621 (2023).
- Gao, Mingqi, et al. “Human-like summarization evaluation with chatgpt.” arXiv preprint arXiv:2304.02554 (2023).
- Liu, Yixin, et al. “BRIO: Bringing order to abstractive summarization.” arXiv preprint arXiv:2203.16804 (2022).
- Fabbri, Alexander R., et al. “Summeval: Re-evaluating summarization evaluation.” Transactions of the Association for Computational Linguistics 9 (2021): 391-409.
- Goyal, Tanya, Junyi Jessy Li, and Greg Durrett. “News summarization and evaluation in the era of gpt-3.” arXiv preprint arXiv:2209.12356 (2022).
- Hu, Yebowen, et al. “Analyzing Influential Factors in Human Preference Judgments via GPT-4.” arXiv preprint arXiv:2305.14702 (2023).

4.5 RESOURCES

- https://python.langchain.com/docs/get_started/introduction
- https://openai.com/
- https://huggingface.co/
- https://en.wikipedia.org/wiki/Alexander_the_Great
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- https://realpython.com/
- https://medium.com/