TEXT SUMMARIZATION - A STUDY ON TEXT SUMMERIZATION WITH PRETRAINED ENCODERS USING PYTHON

Y.S.LIKITH RAO
Student, Business Analytics,
Siva Sivani Institute of Management, Hyderabad, Telangana, India

ABSTRACT

In this age, where an enormous amount of information is available on the Web, it is most basic to supply the advanced mechanism to remove the data quickly and most profitably. It is uncommonly troublesome for human beings to physically extricate the outline of expansive records of content. There is a huge number of textual content, image content available on the Internet. so there's an issue, looking for significant documents from the number of reports accessible, and retaining significant data from it. To solve these issues, the programmed content summarization is exceptionally much necessary. Text Summarization (TS) is the method of recognizing the foremost imperative important data in a report or set of related reports and abstracting them into a shorter form protecting its general implications or meanings of the sentences. It is in this way we are ready to summarize the substance so that it gets simpler to ingest the information, keeping up the substance, and understanding the information.

Keywords

Text summarization, Systematic review , Trend, Feature, Problem, Method

I. INTRODUCTION

After completion of my internship at Intelyica as a Data Analyst Intern which is went 7 weeks where I learned Python: Zero to Pandas and machine learning, I'm now finally able to do my IIP project that's associated with Automatic Text summarization. So here I am going to use a football Wikipedia. So, I'll reduce the Wikipedia huge phrases into short sentences using Python.

The reason why I select Text Summarization is to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. There is an enormous amount of textual material, and it is only growing every single day.

II. LITERATURE REVIEW

In 2019, Oguzhan Tas and Farzad Kiyani [1] discussed about types of summarization methods which might be used in a system to produce summary. Text summarization methods can be classified into extractive and abstractive summarization. Especially, they focused on

extractive summarization methods. Automatic Text Summarization is a popular research area and attracts a lot of attentions from different science branches.

Mehdi Allahyari, Elizabeth D. Trippe, Mehdi Assefi , Juan B. Gutierrez, Saeid Safaei [2] emphasized various extractive approaches for single and multi-document summarization. They described some of the most extensively used methods such as topic representation approaches, frequency-driven methods, graph-based and machine learning techniques.

S

Vishal Gupta, Gurpreet Singh [3] concentrated on extractive summarization methods. An extractive summary is selection of important sentences from the original text. The importance of sentences is decided based on statistical and linguistic features of sentences.

Josef Steinberger, Karel Je˘zek [4] introduced metrics, which are based on latent semantic analysis that can capture the main topics of an article. They experimentally compared the approach with state-of-the-art 1022 J. Steinberger, K. Je˘zek evaluation measures. They also demonstrated that the system ranking provided and correlates well with the human ranking when comparing summaries with abstracts.

Yang Liu and Mirella Lapata [5] showcased how pretrained BERT can be usefully applied in text summarization. We introduced a novel document-level encoder and proposed a general framework for both abstractive and extractive summarization.

III. METHODOLOGY

EXTRACTIVE TEXT SUMMARIZATION METHOD

- ✓ The Extractive based summarization method selects informative sentences from the document as they exactly appear in source based on specific criteria to form summary.
- ✓ The main challenge before extractive summarization is to decide which sentences from the input document is significant and likely to be included in the summary.
- ✓ For this task, sentence scoring is employed based on features of sentences. It first, assigns a score to each sentence based on feature then rank sentences according to their score. Sentences with the highest score are likely to be included in final summary.

IV. ANALYSIS

- URL extraction using python
- Beautifulsoup package is used for processing HTML content
- LexRankSummarizer to Rank the words
- segmentation and the removal of unnecessary words

- Export data
- Data Preprocessing
- Secondary data
- Conclusion.

Here We are importing required packages.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import nltk

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from webdriver_manager.chrome import ChromeDriverManager
```

This is the webpage url which we are opening and parsing(scraping)

```
url = https://en.wikipedia.org/wiki/Football
```

we are using chrome driver to run selenium script - which means opening new google chrome and webpage in that

```
s=Service(ChromeDriverManager().install())
```

we are initialising the web browser here (chrome)

```
browser = webdriver.Chrome(service=s)

browser.maximize_window()
```

using the browser we are opening the url which we specified

```
browser.get(url)
```

Here we are selecting the content from the <body> tag from the html webpage which is opened in the chrome

```
parent_tags = browser.find_element(By.CSS_SELECTOR,'div')

div_tag = browser.find_element_by_css_selector('body')
```

here we are selecting the <p> paragraph tags which are inside the body tag

```
para_tags = div_tag.find_elements_by_css_selector('p')
```

Beautifulsoup package is used for processing HTML content

```
from bs4 import BeautifulSoup
```

Here we are preparing all the sentences

```
sentences = '    '

for para in para_tags:
```

After running the above BeautifulSoup function the above paratext data which was in HTML format will be converted to normal text

```
    para_text = para.get_attribute('innerHTML')

    sentences = sentences+ ' ' + para_text

    sentences = BeautifulSoup(sentences, 'html.parser').text

from sumy.parsers.plaintext import PlaintextParser

from sumy.nlp.tokenizers import Tokenizer

from sumy.summarizers.lex_rank import LexRankSummarizer
```

we are converting/parsing/diving the sentences into small tokens (words)

```
parser = PlaintextParser(sentences, Tokenizer('english'))
```

we are using LexRankSummarizer to Rank the words

```
summarizer = LexRankSummarizer()
```

we are parsing the summarized text content which we prepared from html webpage

```
summarized_sentences = summarizer(parser.document, 16)
```

Here we are printing the summarized content which was earlier cleaned from all the unnnesasary html tags

```
for sentence in summarized_sentences:
    print(sentence)
    print('----------------------')
```

V. FINDINGS

The increasing growth of the Internet has made a huge amount of information available. It is difficult for humans to summarize large amounts of text. Thus, there is an immense need for automatic summarization tools in this age of information overload

According to the analysis there are 500 paragraphs which are very long and cant be read in short time with this help it can make it down into 50 paragraphs with short and coherent way to get easily understand

VI. CONCLUSION

In conclusion, the capabilities of the Text Summarization and the power of the Python programming language in implementing an intended network data analysis cannot be overstated.

Through this project, I was able to analyze the Wikipedia content into short paras. The system was built with Python and the implemented Python libraries include, NumPy, Pandas, Matplotlib. we have presented a technical background for document summarization.

This paper has also discussed several challenges as well as surveys of the existing summarization methods In the topic of text summarization research, future work that can be done includes solving feature problems, namely determining the most appropriate features to use in summarizing by selecting features, discovering new features, optimizing commonly used features, feature engineering, using features for semantics, linguistic features, finding features to produce coherence sentences, and add grammatical features.

Preprocessing by the problem dataset using appropriate stemming, besides, to stop word removal and tokenizing, namely to categorize word classes, such as nouns, verbs, adjectives, etc. iii) For extractive summaries, collaborating statistical techniques, fuzzy-based techniques, and machine learning are very challenging to try.

Automatic text summarization is an exciting research area with several applications on the industry. By condensing large quantities of information into short, summarization can aid many downstream applications such as creating news digests, report generation, news summarization, and headline generation. There are two prominent types of summarization algorithms.

VII. REFERENCES

- (Allahyari et al., 2017)Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., D., E., B., J., & Kochut, K. (2017). Text Summarization Techniques: A Brief Survey. International Journal of Advanced Computer Science and Applications, 8(10). https://doi.org/10.14569/ijacsa.2017.081052
- Steinberger, J., & Ježek, K. (2009). Evaluation measures for text summarization. Computing and Informatics, 28(2), 251–275.
- (Kiyani & Tas, 2017)Kiyani, F., & Tas, O. (2017). A survey automatic text summarization. Pressacademia, 5(1), 205–213