Capstone: Topic and Text Analysis for Frequented Websites

Josiah Teh, Pai Hwai, Ellen Jo

James Cook University Singapore

MA3831: Natural Language Processing

Lecturer: Dr. Eric Tham

07/01/2023

Table of Contents

Introduction 4

Web Scaping 5

Data Collection 5

Content Layout 5

Main Program 5

Code Explanation: 6

Methodology / Approach 6

Data Harvest Summary 6

Topic Selection 7

Literature Review 7

Rational Selection 7

Data Pre-processing 7

Justifications and Specifications of Hyperparameters 8

Evaluation 8

Text Summarisation 9

Literature Review 9

Rational Selection 10

Data Pre-processing 10

Evaluation 11

This Page is Left Intentionally Blank

Introduction

Currently, information has rapidly transitioned from being conveyed physically, as a means of newspaper and textbooks, to now being mainly delivered digitally. In this age of information, the internet can be considered as a massive, central library that connects information providers around the world. With this exponential increase of information, students may find it difficult to narrow down and determine what kind of articles, research proposals and forum discussions are needed to further explore their respective fields. Longer works, such as dissertations, theses, and research papers, have an abstract that report the aims and summary of its work, allowing the user to determine whether the work is in line with what is being researched. More common articles, such as newspapers, lack these sections, with more emphasis on catching attention via an interesting headline.

According to Statista in 2022, the market for natural language processing (NLP) is rapidly expanding, projected to be worth over $43 billion by the year 2025. NLP, which can be summarily described as how computer systems developed in the field of artificial intelligence are able to understand human languages, has opened massive opportunities, such as allowing medical facilities to determine if a patient has a disease based on his speech, or even allowing businesses to determine the sentiment of their customers related to their products. On a more personal level, NLP has been used by email providers to detect and filter out spam emails by analysing the text as they pass through the servers, allowing for their removal before they can even reach you.

In this report, we will be covering how we used NLP to take our problem statement: “With the arrival of the information age, information technology has seen an exponential increase of use, and as a result has extracted an unforeseen amount of information, The task in which a student needs to navigate through an extract meanings from this massive amount has become a challenging task.” The three NLP tasks chosen to help with this are web scraping, topic selection, and text summarisation. Webs scraping, which is the process of extracting relevant data from a targeted website, will allow the user to retrieve all useful information easily. Topic analysis, the extraction of a meaning from a text, is resulted by identifying the most important

themes and topics in their respective articles. Text summarisation, is the process in which lengthy text is broken down into paragraphs and sentences, allowing the user to decide if the article selected is worth diving in deeper.

Web Scaping

Web Scraping refers to the extraction of data from a website, which is collected and exported into a format that is more useful for the user. Web scraping can be done on the entire site or specified data for one or multiple URLs. For our portion, we will be scraping only specified data and on multiple URLs; 30 to be exact.

Data Collection

The data collected for web scraping comes from 3 primary websites: Reuters, Towards Data Science, and IBM. We deemed this selection to be appropriate since they're websites frequented by data science students.

Content Layout

- Import libraries
- store URLs
- Extend stored data into a single Collection List
- Main program for scraping and parsing
- Storing DataFrame
- Copying DataFrame into a csv

Main Program

The layout of the main program is storing parsed data into a list of lists. We use a for loop to iterate the program for all 30 URL webpages. Illustrating in simpler terms, we store our scraped data into named variables and store them in a sections list, which will be the list we access for the data.

We also name headers in the inner bracket of the collection list. The data are stored as 3 observations: Title, Body, and URL. We access the list using indexing to create a dataframe

named df. After roughly assuring the data output looks correct by calling head() and tail(), we copy this dataframe into a csv file named research_paper.csv.

```
# Main Program
collection = [["title", "body text", "url"]]

for i in allPageList:
    page = requests.get(f"{i}")
    soup = BeautifulSoup(page.text, "html.parser")
    title = soup.find("h1").text # get title
    all_text = soup.find("p").text # get body
    urls = str(f"{i}") # also get the urls


    section = [title, all_text, urls]
    collection.append(section)
df = pd.DataFrame(collection[1:], columns=collection[0])
```

Figure 1: Code block for main function to extract text

Code Explanation:

We are only scrapping the url, title and body for this program. Using BeautifulSoup and html.parser, we can access the title, and body cleanly. By just inspecting the webpage's title and body text, we know that it's stored as "h1" and "p" in the html format. Which is how we access the data and store it as title and all_text respectively. We also store the URLs as string for formatting.

Methodology / Approach

When choosing our primary websites, we inspected the website's web page language. All three websites were in the HTML language. With this information, we have a better idea on how to parse our data extraction. The main libraries we used in the main program are requests, BeautifulSoup, and pandas. We used requests library to get requests to the url and BeautifulSoup to pull texts from the webpages and parse using html.parser. BeautifulSoup was an appropriate library for our choice because it is a simple and commonly used library to pull text and parse for HTML and XML files. Pandas is used to create the data frame, as we have learned in previous classes.

Data Harvest Summary

We can now pull up the csv file and see our 3 observations of title, body text and URL. The title strings are clean, and the body text includes some time stamps and stop words. The

URLs strings are clean as well and all 30 webpages are present in the csv. This completes our web scraping portion, and the csv file and data frame will be used later in the next portion.

Topic Selection

Literature Review

Text summarization is a Natural Language Processing (NLP) task that aims to condense a large amount of text into a shorter, more concise version while retaining the main information and meaning. There are several types of text summarization, including extractive and abstractive summarization. Extractive summarization involves selecting and extracting important sentences or phrases from the original text, while abstractive summarization involves generating a new summary by understanding and paraphrasing the original text (Miroshnyk, 2022). Topic modelling is an unsupervised classification model that finds and groups items without having the items labelled, which is like numeric clustering. In this model, Latent Dirichlet allocation (LDA) method will be implemented as it is the most common method for fitting a topic model and mostly used for natural language pro-cessing, text mining, and social media analysis, information retrieval (Jelodar et al, 2017). LDA can estimate both mixture of associated words in topic and mixture of topics correlating with the document.

Rational Selection

The LDA method is the best method for fitting our topic model because it is popular among users for topic modelling and text mining. The method does not take long to compute and is good with big data. By using this topic modelling technique, it is possible to determine the key points being made in the documents without having to go through every detail in each one (Nair, 2022).

Data Pre-processing

Pre-processing the data starts with importing important library. For this model, we will use genism NLTK, pandas, and pyLDA for visualization. For data wrangling with regular expression, we will remove punctuation, convert uppercase to lowercase. We will have a clean data to attach to the dataset frame. Next comes the simple pre-processing with genism, NLTK, where we tokenize the words, omit stop words as well as the punctuations. Then we create a document term matrix and dictionary.

Justifications and Specifications of Hyperparameters

There are some specifications that are introduced into the model that is the number of topic and the number of documents for training. For this model, we used a total of 30 documents and 3 topics due to the limited number of documents we scrap. However, the initial amount of topic was set to 5 and when the model was tested, the result shows that the model can only detect 4 topics. Because of that, we changed from 5 to 3 topics as we do not want too many or too little amount of topic that can be detect by this simple model. Normally, a topic modelling should be trained with at least 1000 documents for higher accuracy. But due to the lack of documents and computing power, we will only build a simple model to run with 30 documents in a corpus.

Evaluation

We start by importing the scrap document dataset and essential library. Then proceed by cleaning the data to make it easier to process by removing irrelevant information, punctuations and uppercases. With genism nltk, we can do a simple pre-processing where the datasets will have their words from the documents tokenized. Because there is an abundance of unnecessary words such as stop words in the documents, it can be easily omitted with nltk library. We can then lemmatize the words so that multiple words of similar definition can be grouped together. The tokens will be appended to list as document which will be used to make a dictionary and document term matrix. Before building the LDA model, the number of topics must be decided to represent the group of topics the model will detect. This process is quite challenging as there is no formula to decide the right number. Thus, we start with 5 topics and run the model with it. In the end, we decide to put 3 topics as the model can only detect 4 topics at most. Based on the result that we visualise with pyLDAvis from our document, we found that topic 1 has the most proportion of documents containing the topic and ITC being the most relevant out of the other words.

To conclude, the model's performance is not bad. But because the model is an unsupervised learning, the result will be highly dependent on the user's choice for number of topics. Finally, the limited amount of data we can get are far from enough which means the results may be evenly distributed or biased.

Text Summarisation

Literature Review

Simply put, the goal of text summarisation is to perform a reductive transformation of source text to produce a summary text through methods such as content selection and generalisation of what it determines is important within the source (Jones, 1999). This process is essential to allow the user to save time and resources by allowing the discovery of specific information within the documents. Though it was developed in 1950s (Luhn, 1958), text summarisation has gone through many overhauls within the 2000's, being exposed to a wide range of paradigms and techniques. However, even after all this, text summarisation still proves to pose many challenges. Issues, such as data redundancy and irregular sentence ordering, become even more prominent when approaching multi-document summarisation (Goldstein et al, 2000).

Regarding the summary of text summarisation, one must note the definition is not strict. This definition varies on several factors, such as the main intention of the summary, what are the context factors that influence the summaries and the targeted audience types (Jones, 1999). While taxonomies have been the main driving force behind these models, entity level approaches, where the model uses documents to create entities and map their relationships to each other, have gotten more popular in the early 2000s. These approaches consider patterns of connectivity within the document, such as the relationship of the words to each other and the distance between text units.

Text summarisation has since transited to machine learning approaches, with the very first model being the Hidden Markov Models. Unlike the previous methods, a wide range of machine learning techniques can be applied for text summarisation. In the more recent years, supervised and semi-supervised approaches have been used to detect relevant information via a query-focused summary method (Fuentes et al, 2007). Overall, the main advantage of using machine learning for text summarisation is that performance can be tested easily over a number of different measures. These methods, however, require massive corpuses to function effectively (Lloret & Palomar, 2009).

After identifying sections that were repeated among articles from the same domain, regular expression was applied, followed by applicable list indexing to make sure the summarisation focus wasn't changed. Finally, these cleaned articles were run through a function that split the sentences in articles with a maximum of 500 words. The reason for this is because T5-small, the model we selected has issues when indexing chunks more than 512 words.

Evaluation

As we tested our summariser, it could be noted that while almost all summaries were relevant to the title, certain articles were classified better than others.

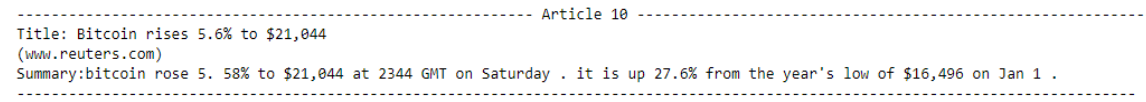

Figure 4: Summarisation for article 10

Above is an example of a well classified article, in which all the key point of the data were summarised in one sentence.

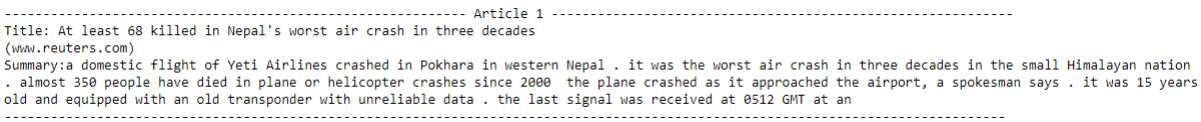

Figure 5: Summarisation for article 1

Another article which was well summarised but seems to have issues with sentence structure at the end of the sentence.

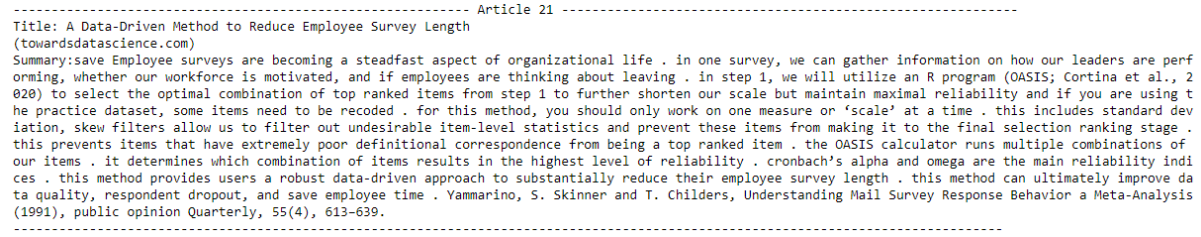

Figure 6: Summarisation for article 21

Article in which summary is related to the topic, but content is rather irrelevant, containing citations and book titles, and having major issues with forming proper sentence structure.

References

Fuentes M, Alfonseca E, Rodríguez H (2007) Support vector machines for query-focused summarization trained and evaluated on pyramid data. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions. pp 57–60

Goldstein J, Mittal V, Carbonell J, Kantrowitz M (2000) Multi-document summarization by sentence extraction. In: NAACL-ANLP 2000 workshop on automatic Summarization. pp. 40–48

Lloret E, Balahur A, Palomar M, Montoyo A (2009) Towards building a competitive opinion summarization system: challenges and keys. In: Proceedings of the NAACL. Student Research Workshop and Doctoral Consortium. pp 72–77

Luhn HP (1958) The automatic creation of literature abstracts. In: Advances in automatic text summarization. pp 15–22

Miroshnyk, O. (2022). What is summarizing?. One AI. https://www.oneai.com/learn/text-summarizer

Nair, A. (2022). Topic Modeling With Latent Dirichlet Allocation: An overview and implementation of a popular modeling technique in NLP. Towards Data Science. https://towardsdatascience.com/topic-modeling-with-latent-dirichlet-allocation-ea3ebb2be9f4

Spärck Jones K (1999) Automatic summarizing: factors and directions. In: Advances in automatic text summarization. pp 1–14