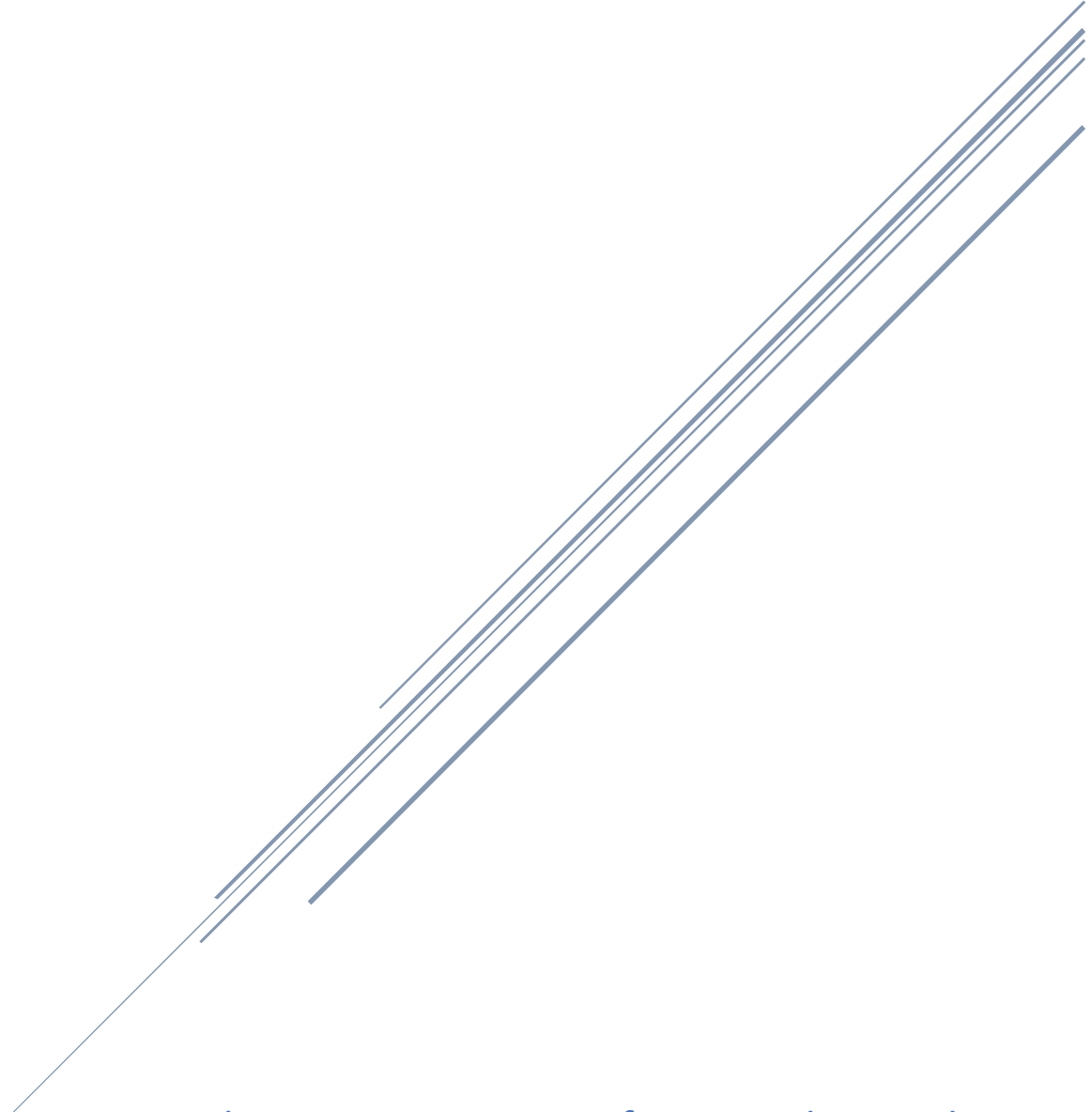


# COMP 4601 FINAL PROJECT – WIKIPEDIA ARTICLE SUMMARIZER

Murad Berhanu - 100996375



Carleton University – Professor Anthony White  
COMP4601 Intelligent Web-Based Systems

## **Abstract**

My final project is an application that uses text summarization techniques to summarize Wikipedia articles. By going to a certain URL that contains part of a Wikipedia URL (for example, to summarize [https://en.wikipedia.org/wiki/Automatic\\_summarization](https://en.wikipedia.org/wiki/Automatic_summarization) you would keep the ending part and go to [http://localhost:8080/TextSummarizerSystem/rest/ts/Automatic\\_summarization/](http://localhost:8080/TextSummarizerSystem/rest/ts/Automatic_summarization/)), an HTML page will be displayed by the system with a link to the original article with the summary under it. The summary length can be modified by the user by adding a “/” followed by a number to the end of the request URL. The application is developed as a RESTful web application, and the only external code comes from using jsoup. No libraries such as Stanford CoreNLP [4] were used. The code is based on Babluki’s text summarization algorithm [1].

## **Introduction**

With the huge amount of information on the internet and the fast-paced society we have today, people need to be able to see information that is relevant in a quick manner. Everything these days has a short summary or description to try to catch a user’s attention or to let them consume information quickly. Everything from Google search results, movie synopses, video descriptions and news articles has been compressed to provide easy to consume information. In a world with so much information available at the click of a button, there needs to be a way for relevant information to be delivered quickly.

This is where automatic text summarization comes in. Rather than reading an entire document or article, text summarization algorithms attempt to deliver only the information that is most important. In an attempt to develop a useful text summarization system, I have developed a RESTful web application that summarizes Wikipedia articles. Wikipedia articles can be ridiculously long sometimes and contain a wide variety of information on a single subject, so it seems to be a good candidate for a summarization system to be used for.

This report consists of various sections that will follow. These sections are: the background, related work, methodology, discussion, conclusion, and future work. The background section describes the background information needed for this project. The related work section includes descriptions of various works related to this project. The methodology section contains a summary of how the project is designed and the resulting functionality. The discussion section details what worked, what failed, and the strengths and weaknesses of the project. The conclusion section simply concludes the report. Finally, the future work section describes how I think the project could be improved and expanded on in the future.

## **Background**

As the project is based on text summarization, the ideas behind text summarization need to be understood before reading the rest of the paper. Text summarization is an application of natural language processing (NLP). Text summarization is the idea of using computers to automatically summarize text and provide a short summary which conveys the essence of an article, document, or other text. This is done to help find relevant information quickly. In general, a summary should contain a significant portion of the information in the original text, while being no more than half the length of the original text [6].

There are two types of summaries: extractive summaries and abstractive summaries. Extractive summaries are simpler, and are created by taking parts of the original text and using those directly in the summary. Abstractive summaries are more complex and are created by rephrasing the original text, similar to the summary on the back of a book or a movie review [6]. This project uses extractive summarizing techniques, which are by far more widely used today.

Summarization techniques can also be supervised or unsupervised [6]. This project uses unsupervised techniques, which does not label responses. Namely, TextRank. TextRank is an algorithm similar to PageRank that ranks sentences by importance. This will be described in more detail in the “Methodology” section.

Other things that should be understood are RESTful web applications, jsoup HTML parsing, tomcat servers, and basic Java.

## **Related Work**

There have been many works in text summarization related to my project. The most important example is the **Google search results**. Google search results provide a summary of the articles that are shown, although they are more focused on the specific search terms that you have typed in rather than providing a summary of the entire webpage.

Another relevant work is the **Stanford CoreNLP** (Natural Language Processor) [4]. Natural language processing is a massive field and goes far beyond text summarization. Stanford CoreNLP is a library that provides a set of human language technology tools. Among the various tools, it provides a text summarization tool.

**Babluki’s text summarization algorithm** [1] is another directly related work to my project. The project is based on implementing Babluki’s text summarization algorithm to provide a summary of the Wikipedia articles.

## **Methodology**

The application is developed using Java as a RESTful web application. The application contains three main components: input, processing, and output.

## Input

The program starts by the user first going to a request URL. The user inputs a URL that contains the Wikipedia page they want to summarize. The base URL is <http://localhost:8080/TextSummarizerSystem/rest/ts/> and you add what you want to add after the backslash. For example, to summarize the page <https://en.wikipedia.org/wiki/Ottawa>, you would need to add the portion after “wiki/” to the URL and make <http://localhost:8080/TextSummarizerSystem/rest/ts/Ottawa>. The user can also add a desired maximum number of sentences for the summary to the URL. The default is 15, but adding a “/” followed by a number to the end of the URL defines the maximum number. Wikipedia has a standardized URL format so obtaining the right URL is simple. This calls a RESTful GET request which calls the initialization function.

The initialization function is used to take the entire webpage and divide it into sentences and paragraphs. This is done using jsoup. The function first constructs the proper Wikipedia URL by parsing the localhost URL, and then uses jsoup to connect to the webpage. A jsoup document is created and the parsing begins. It modifies certain elements (for example, removing superscript tags and unwrapping anchors), and then it selects only the paragraph (“<p>”) elements from the page and returns that as a single string. This string is the text that will be summarized. For each paragraph element, a paragraph object is created. Paragraph objects contain sentence objects. These sentence objects are created by using a regular expression to split the paragraph into sentences. Sentences and paragraphs are numbered to be used later. This concludes the input portion of the application.

## Processing

With the input finished, the processing can begin. This is done using a modified version of Babluki’s algorithm. The first step is to create a similarity matrix. Similarities (intersections) between all sentences are calculated and stored in a matrix. The similarity value is the number of words two sentences have in common divided by the number of words in the two sentences combined divided by 2. In java, this is “noOfCommonWords(sentence1, sentence2) / ((double)( sentence1.noOfWords + sentence2.noOfWords) /2);” The score for each sentence is then calculated using the matrix and stored in each sentence object. The score is calculated by adding up all the similarities a given sentence has between all other sentences in the matrix.

Once the scores are calculated, the summary can be formulated. First, only a certain number of sentences can be selected from each paragraph. The number is proportional to the size of the paragraph. This is to ensure a more even spread of the summary. Only a certain number of the most relevant sentences will be selected from each paragraph and added to the summary. The sentences are then sorted by their number stored in the input stage to ensure proper ordering.

## Output

Once the initial summary is generated, the summary will be shortened even more to only contain the user-specified number of sentences. It will remove the lowest scored sentences until the maximum number is reached. This summary is then used in the original RESTful GET request to generate the HTML page for the summary that is displayed.

## Design Decisions

There were a few decisions I had to make in terms of the design. Firstly, I decided to use Wikipedia pages as documents to summarize. I chose this since it contains a large number of documents in a standardized format. The URLs were also easy to use as they are human-readable compared to, for example, a BBC (bbc.com) article which is generally just seemingly random numbers.

Another big decision I had to make was whether to keep stop words or remove them when generating the similarity matrix. In Babluki's original algorithm, he does nothing about removing stop words. I wondered why, so I compared my results when I removed them versus when I didn't. Here are 3 sample pages when running my program on the Wikipedia page "Ottawa".

### Summary of wikipedia page: [Ottawa](#)

1. Ottawa borders Gatineau, Quebec, and forms the core of the Ottawa-Gatineau census metropolitan area (CMA) and the National Capital Region (NCR). As of 2016, Ottawa had a city population of 934,243 and a metropolitan population of 1,323,783 making it the fourth-largest city and the fifth-largest CMA in Canada
2. The House of Commons and Senate was temporarily relocated to the then recently constructed Victoria Memorial Museum, now the Canadian Museum of Nature until the completion of the new Centre Block in 1922, the centrepiece of which is a dominant Gothic revival styled structure known as the Peace Tower
3. Greber's plan included the creation of the National Capital Greenbelt, the Parkway, the Queensway highway system, the relocation of downtown Union Station (now the Government Conference Centre) to the suburbs, the removal of the street car system, the decentralization of selected government offices, the relocation of industries and removal of substandard housing from the downtown and the creation of the Rideau Canal and Ottawa River pathways to name just a few of its recommendations
4. Ottawa is on the south bank of the Ottawa River and contains the mouths of the Rideau River and Rideau Canal
5. It was able to bypass the unnavigable sections of the Cataraqui and Rideau rivers and various small lakes along the waterway due to flooding techniques and the construction of 47 water transport locks. The Rideau River got its name from early French explorers who thought the waterfalls at the point where the Rideau River empties into the Ottawa River resembled a "curtain". Hence they began naming the falls and river "rideau" which is the French equivalent of the English word for curtain
6. Ottawa is bounded on the east by the United Counties of Prescott and Russell; by Renfrew County and Lanark County in the west; on the south by the United Counties of Leeds and Grenville and the United Counties of Stormont, Dundas and Glengarry; and on the north by the Regional County Municipality of Les Collines-de-l'Outaouais and the City of Gatineau
7. The main suburban area extends a considerable distance to the east, west and south of the centre, and it includes the former cities of Gloucester, Nepean and Vanier, the former village of Rockcliffe Park (a high-income neighbourhood which is adjacent to the Prime Minister's official residence at 24 Sussex and the Governor General's residence), and the communities of Blackburn Hamlet and Orléans
8. The city had a population density of 334.8/km (867/sq mi) in 2016, while the CMA had a population density of 195.6/km (507/sq mi). It is the second-largest city in Ontario, fourth-largest city in the country, and the fourth-largest CMA in the country
9. As of 2015, the region of Ottawa-Gatineau has the sixth highest total household income of all Canadian metropolitan areas (\$82,052). The median household income after taxes is \$73,745 which is higher than the national median of \$61,348. The unemployment rate in Ottawa in 2016 was 7.2%, lower than the national rate of 7.7%. In 2019 Mercer ranks Ottawa with the third highest quality of living of any Canadian city, and 19th highest in the world
10. The city is also home to the Canada Agriculture Museum, the Canada Aviation and Space Museum, the Canada Science and Technology Museum, Billings Estate Museum, Bytown Museum, Canadian Museum of Contemporary Photography, the Bank of Canada Museum, and the Portrait Gallery of Canada

[Figure 1 – without removing stop words]

## Summary of wikipedia page: [Ottawa](#)

1. He also laid out the streets of the town and created two distinct neighbourhoods named "Upper Town" west of the canal and "Lower Town" east of the canal
2. Greber's plan included the creation of the National Capital Greenbelt, the Parkway, the Queensway highway system, the relocation of downtown Union Station (now the Government Conference Centre) to the suburbs, the removal of the street car system, the decentralization of selected government offices, the relocation of industries and removal of substandard housing from the downtown and the creation of the Rideau Canal and Ottawa River pathways to name just a few of its recommendations
3. Ottawa is on the south bank of the Ottawa River and contains the mouths of the Rideau River and Rideau Canal
4. Ottawa sits at the confluence of three major rivers: the Ottawa River, the Gatineau River and the Rideau River
5. It was able to bypass the unnavigable sections of the Cataraqui and Rideau rivers and various small lakes along the waterway due to flooding techniques and the construction of 47 water transport locks. The Rideau River got its name from early French explorers who thought the waterfalls at the point where the Rideau River empties into the Ottawa River resembled a "curtain". Hence they began naming the falls and river "rideau" which is the French equivalent of the English word for curtain
6. Across the Ottawa River, which forms the border between Ontario and Quebec, lies the city of Gatineau, itself the result of amalgamation of the former Quebec cities of Hull and Aylmer together with Gatineau
7. Ottawa is bounded on the east by the United Counties of Prescott and Russell; by Renfrew County and Lanark County in the west; on the south by the United Counties of Leeds and Grenville and the United Counties of Stormont, Dundas and Glengarry; and on the north by the Regional County Municipality of Les Collines-de-l'Outaouais and the City of Gatineau
8. The main suburban area extends a considerable distance to the east, west and south of the centre, and it includes the former cities of Gloucester, Nepean and Vanier, the former village of Rockcliffe Park (a high-income neighbourhood which is adjacent to the Prime Minister's official residence at 24 Sussex and the Governor General's residence), and the communities of Blackburn Hamlet and Orléans
9. The city had a population density of 334.8/km (867/sq mi) in 2016, while the CMA had a population density of 195.6/km (507/sq mi). It is the second-largest city in Ontario, fourth-largest city in the country, and the fourth-largest CMA in the country
10. The city is also home to the Canada Agriculture Museum, the Canada Aviation and Space Museum, the Canada Science and Technology Museum, Billings Estate Museum, Bytown Museum, Canadian Museum of Contemporary Photography, the Bank of Canada Museum, and the Portrait Gallery of Canada

[Figure 2 – removing stop words with a random list found online]

## Summary of wikipedia page: [Ottawa](#)

1. Ottawa borders Gatineau, Quebec, and forms the core of the Ottawa-Gatineau census metropolitan area (CMA) and the National Capital Region (NCR). As of 2016, Ottawa had a city population of 934,243 and a metropolitan population of 1,323,783 making it the fourth-largest city and the fifth-largest CMA in Canada
2. Ottawa has the most educated population among Canadian cities and is home to a number of post-secondary, research, and cultural institutions, including the National Arts Centre, the National Gallery, and numerous national museums
3. Secondly, Ottawa was approximately midway between Toronto and Kingston (in Canada West) and Montreal and Quebec City (in Canada East). Additionally, despite Ottawa's regional isolation, it had seasonal water transportation access to Montreal over the Ottawa River and to Kingston via the Rideau Waterway
4. Rail lines built in 1854 connected Ottawa to areas south and to the transcontinental rail network via Hull and Lachute, Quebec in 1886. The original Parliament buildings which included the Centre, East and West Blocks were constructed between 1859 and 1866 in the Gothic Revival style
5. It also spread across the Ottawa River and destroyed about one fifth of Ottawa from the Lebreton Flats south to Booth Street and down to Dow's Lake
6. Ottawa is on the south bank of the Ottawa River and contains the mouths of the Rideau River and Rideau Canal
7. Across the canal to the west lies Centretown and Downtown Ottawa, which is the city's financial and commercial hub and home to the Parliament of Canada and numerous federal government department headquarters, notably the Privy Council Office
8. Ottawa is bounded on the east by the United Counties of Prescott and Russell; by Renfrew County and Lanark County in the west; on the south by the United Counties of Leeds and Grenville and the United Counties of Stormont, Dundas and Glengarry; and on the north by the Regional County Municipality of Les Collines-de-l'Outaouais and the City of Gatineau
9. The main suburban area extends a considerable distance to the east, west and south of the centre, and it includes the former cities of Gloucester, Nepean and Vanier, the former village of Rockcliffe Park (a high-income neighbourhood which is adjacent to the Prime Minister's official residence at 24 Sussex and the Governor General's residence), and the communities of Blackburn Hamlet and Orléans
10. Three main daily local newspapers are printed in Ottawa: two English newspapers, the Ottawa Citizen established as the Bytown Packet in 1845 and the Ottawa Sun, and one French newspaper, Le Droit. Multiple Canadian television broadcast networks and systems, and an extensive number of radio stations, broadcast in both English and French

[Figure 3 – removing stop words with Python's NLTK stop words - [3] ]

When comparing those three pages, they have a few sentences in common. Between Figure 1 (no stop words removed) and Figure 2 (stop words removed), Figure 1 would be the better summarization in my opinion, which was surprising since I thought that would be a definite way to improve Babluki's algorithm. Figure 3, which uses a standardized library's list of stop words, was an improvement over Figure 2. I would say that Figure 1 and Figure 3 gave comparable summaries, even though they were slightly different. So I concluded that there was no noticeable improvement given by removing stop words.

This is further shown in other examples, such as summarizing the Wikipedia page for automatic text summarization. The summary without removing stop words contained the sentence "Automatic summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content" as the first sentence, while the summary



that removed stop words did not contain this sentence or any other that was as seemingly important, although they did have a few other sentences in common.

## **Evaluating the Summary**

It was difficult to determine whether a good summary was being produced using any sort of standardized techniques. Using the Rouge-1 metric was too difficult to implement in this project. The only clear way was to use human judgement and summarize articles on topics that I had a fair understanding of. For example, I know that the sentence “As of 2016, Ottawa had a city population of 934,243 and a metropolitan population of 1,323,783 making it the fourth-largest city and the fifth-largest CMA in Canada” would be a useful piece of information to have in a summary, but “A new Central Post Office (now the Privy Council of Canada) was constructed in 1939 beside the War Memorial because the original post office building on the proposed Confederation Square grounds had to be demolished” is not exactly crucial information.

Another technique I used was comparing my summaries to other summarizers available online. I used websites such as textsummarization.net [5] to see if mine looked comparable to theirs, or if the summaries seemed better or worse. That is the difficult thing about natural language processing, there is no clear way to interpret results without using humans.

I also had to try summarizing many different articles to try and find as many formatting issues as possible to improve my parsing of the information.

Testing different lists of stop words was also important, as they produced completely different summaries.

## **Discussion**

### **Strengths**

I think that my project had a few strengths. In terms of the usability, I think that it was fairly easy to use. The application runs quickly as well, and there are no glaring issues. It was successfully able to get the correct information from the Wikipedia articles, process the information, and provide a summary using Babluki's algorithm that was as long or as short as the user wanted, as well as providing a convenient link to the full Wikipedia article on the page.

### **Weaknesses**

As well as the application worked, there are of course many weaknesses as well. Firstly being that there was no good way to test how good the summarization was in a standardized way other than through human observation. This is of course a problem much bigger than what can be solved in an undergraduate assignment, but it is a

problem nonetheless. Another weakness comes in the usability. While it is simple to use if you know what to do, having to modify the URL directly is not exactly the most user-friendly way to design an application. Also, the user has to know part of the actual Wikipedia URL to be able to use the application. It is fine in many cases since the portion that needs to be known is generally intuitive (e.g. for the Ottawa Wikipedia page, you just need to know the text “Ottawa”), it can be more complicated in some cases (e.g. “Trainspotting\_(film)”, or knowing the difference between “1984” (the year), “Nineteen\_Eighty-Four”(the book), or “1984\_(1956\_film)”(the film based on the book released in the year 1956)).

The application is also fairly ugly to be frank. The user interface is bare-bones and doesn’t look very special. The actual summarization could also probably be improved, as there are sometimes sentences that appear that really do not seem very important.

A big challenge I had was with parsing the HTML into sentences. There are many elements of a page that could affect my parsing such as Wikipedia annotations, words appearing that do not use alphanumeric characters (for example Japanese or Chinese words or other symbols), words or numbers with periods in them, or even typos where there is no space left after a period or annotation at the end of a sentence. I had to make decisions on what errors to allow in my final parsing. If I wanted decimals or words such as “e.g.” to not break sentences, then I had to allow spacing typos between sentences. Some articles, in particular the Ottawa Wikipedia page, had many of those typos compared to others. The results of these typos appear in the Ottawa summary pages in the “Methodology” section.

## **What I Learned**

I learned to appreciate exactly how difficult it is to develop a useful summarization system and how deep the field of natural language processing is. I also learned how user input can completely mess with an algorithm, given how many headaches typos gave me. I have seen how unsanitary the web can be. You may think that websites such as Wikipedia would have standardized formats and clean, edited text. I learned that this is definitely not true, and that a lot of the time the web can be sloppy and not developer-friendly. I learned this fact even further in my original attempts at summarizing websites other than Wikipedia, seeing how websites can have completely different HTML representations that increase the workload and headaches.

## **Conclusions and Future Work**

In the end, I have created an application that provides a summary for Wikipedia articles. User input is obtained from the URL. The information is parsed from the webpages



using jsoup. It uses a modified version of the Babluki algorithm to output the summary to an HTML page obtained from a RESTful web application.

There are many things I believe can be improved in the future. Originally, I wanted the application to be a webpage where you can enter any link and then the system will find the page and then summarize the content (similar to the website in reference [5]) rather than the approach I have where you modify the URL with part of a Wikipedia URL. That turned out to be more work than I expected and I had to abandon that idea for this project. I also think that I could have parsed the content better, as in the parsing issues that I have explained in the discussion section. Improving these issues would also improve the final output of the program if words and sentences are detected properly. Multilingual support could also be added which would introduce another layer of difficulty.

## **References**

- [1] Babluki, S. (2013, April 28). Build your own summary tool! Retrieved from <https://thetokenizer.com/2013/04/28/build-your-own-summary-tool/>
- [2] Build a Text Summarizer in Java. (17AD). Retrieved from <https://www.youtube.com/watch?v=1PXGcUA3m18>
- [3] Natural Language Toolkit. (n.d.). Retrieved from <https://www.nltk.org/>
- [4] Stanford CoreNLP. (n.d.). Retrieved from <https://stanfordnlp.github.io/CoreNLP/>
- [5] Text Summarizer - TextSummarization: Text Summarization Online: Text Summarization Demo: Text Summarization API. (n.d.). Retrieved from <http://textsummarization.net/text-summarizer>
- [6] White, A. (n.d.). COMP4601 Document Summarization. Retrieved from <https://sikaman.dyndns.org:8443/WebSite/rest/site/courses/4601/handouts/13-Text-summarization.pdf>