Automatic Text Summarization of Web Pages Using Commonly Copied Sentences

by
Sagi Behor
Supervisor: Dr. Ilan Kirsh

Thesis

The Academic College of Tel-Aviv Yaffo
School of Computer Science

September 2021

Table of Contents

1 Introduction

Text summarization is the process of creating a short and informative summary of a text document or documents. Automatic text summarization methods are needed to address the growing amount of online text data and help discover and consume relevant information faster. Text summarization was a known problem long before the internet. The seminal papers on automatic summarization were published more than 60 years ago, and since then, the practical need for automatic summarization has become urgent, and numerous papers have been published on the topic. The World Wide Web contains billions of documents and grows exponentially, mainly because of the large amounts of text data created in various social networks, web, and other applications. As a result, there has been a tremendous need to design methods, algorithms, and tools that can effectively process various text applications.

Over the years, many automatic summarization tools have been presented, and each has its limitations [1][2][3]. In this thesis, I present and examine a new automatic extractive text summarization approach. This approach is based on "crowd wisdom" and assumes web users copy important information more frequently. The algorithm splits the text of a web page into sentences, searches every sentence on a search engine, and records result counts. The summary is created from the sentences with the highest search result counts. As the experiments show, this algorithm excels where other algorithms usually have limitations - summarizing short texts without previous knowledge of the topic.

2 In-depth Literature Review

Most tools that provide automatic summarization can be divided into two main types: Abstractive Summarization and Extractive Summarization.

2.1 Abstractive Summarization

Abstractive Summarization methods build a summary by generating new sentences from the information in the original text. Summaries made by humans usually have abstraction values, where sentences in the summary can be new sentences that convey the information from the original article. The automatically generated abstractive summaries are expected to be better and closer to a human-built summary.

Algorithms that try to create abstractive summaries have three main challenges: 1) Compression. The following two challenges come to mind when thinking about abstractive summaries, but to meet the need of every summary, the algorithm's output must be shorter than the original text. The algorithm uses the previous training process, which created a model that recognizes important words by topic and tokenizes connection between words. 2) Sentence Fusion. This stage comes together with the compression stage. The compression stage tells the meaningful information from one or more sentences, and the Fusion technique is required to combine it and discard the repetitive words. 3) Reorganization. A significant challenge for Abstractive summarization tools is to create a coherent and human-readable sequence of sentences. A known way to meet this task is to use sentence templates uniquely created for the summary language [4].

The quality of this summarization type in automatic summaries is still inadequate [5], and one of the main reasons for that is connected to Knowledge Representation and Reasoning subject. Knowledge Representation in Artificial Intelligence (AI) is concerned with how

knowledge can be represented symbolically by reasoning programs. The observation can explain this connection that most text understanding requires grammatical knowledge about the specific language the text is written in and has to incorporate prior knowledge about the text domains. Thus, the inferencing capabilities of knowledge representation languages are crucial for text understanding systems [6]. Knowledge Representation and Reasoning (KRR) has been studied for many years and the significant progress used to develop many disciplines as pragmatics, natural language semantics, linguistics, cognitive psychology, and artificial intelligence (AI). With that being said, Knowledge Representation and Reasoning research will probably never be complete due to the deep conflicts in its base. Cognitive and a metaphysical notion of context are examples of the dichotomies that appear in many context theories, like subjective and objective, internal or external, and more. There is no ground truth that the research can relate to [7].

2.2 Extractive Summarization

The extractive summarization process builds a summary by ranking the importance of every sentence in the original text and generating the summary from the sentences with the highest scores. The required summary length depends on the specific application. Of course, there are many ways to determine how vital each sentence is. The main methods are described below.

2.2.1 Topic-based approaches

An algorithm of this approach assumes it knows what the text is about (the subject). The algorithm is trained with many examples of topics and important words or sentences connected to them. That is the primary key factor when the algorithm decides which score to assign to each sentence. The users do not have to insert the topic into the algorithm manually. Instead, the algorithm can use a different algorithm that automatically finds the topic by the title or the

abstract of the text. This approach is common when summarizing text of more the one document [8].

2.2.2 Graph-based approaches

Graph-based algorithms are relatively new. In this approach, a graph with nodes and edges is generated from the text. If the similarity score of two sentences exceeds a given threshold, there is a connection (an edge) between them. There are various ways to select the best sentences for inclusion in the summary, but the main principle is that sentences with more connections are more likely to be included.

The two most popular algorithms of this approach are LexRank [9] and TextRank [10]. In order to find the semantic similarity, LexRank creates a TF*IDF (term frequency-inverse document frequency) vector for each sentence. The cosine similarity between two corresponding vectors defines the similarity between the two sentences.

On the other hand, TextRank measures semantic similarity based on the number of words two sentences have in common, normalized by the sentences' lengths.

LexRank and TextRank assign scores to the sentences using implementations inspired by the PageRank [11] algorithm.

PageRank (first used by the Google search engine) is an algorithm used primarily for ranking web pages in online search results based on evaluating the probability of users to visit each page. For the calculation process, PageRank examines links between pages. The idea is that a page with more inbound links is more important and should be ranked higher in the search results. PageRank creates a square matrix with n rows and n columns, where n is the number of web pages. Each element of this matrix denotes the probability of a user transitioning from one web page to another. After calculating the links, the algorithm assigns a score for each page, when the page with the highest score should be first in the search engine results

The PageRank terminology is about web pages, but LexRank and TextRank use it to find important sentences. Instead of working on actual links between the web pages, each "link" is a semantic similarity between two sentences. The only difference is that the graph generated by PageRank is a directed graph because each link connects one page to another, when the graph generated by LexRank and TextRank is undirected as the semantic similarity between sentences is bidirectional.

2.2.3 Statistical-based approaches

This approach extracts essential sentences and words from the original text using statistical features. The statistical features are usually connected to the structure of the text and completely independent and separated from the literal and logical content of the text. These techniques do not require any previous linguistic knowledge or complex linguistic processing. Examples of this kind of features are the position of the sentence, the centrality of the sentence (based on similarity to other sentences), relative length of the sentence, the resemblance of the sentence to the title, presence of numerical data in the sentence, or presence of proper noun (name entity) in the sentence. This approach is often integrated with another approach for creating a summary. Because this method is related directly to the text structure, it has limitations on the length and structure of the original text [12].

2.2.4 Machine learning based approaches

Although machine learning algorithms are mostly connected to abstractive summaries, many extractive summarization algorithms that use machine learning have been studied in recent years. These algorithms can be supervised, unsupervised or semi-supervised. In a supervised approach, the algorithm has supervised or trainable summarizers that classify each sentence of the test document either into "in-summary" or "not-in-summary" classes with the help of a training set of documents. With a large amount of labeled or

annotated data for the learning purpose, known supervised algorithms, like Support Vector Machine (SVM), Naïve Bayes classification, Mathematical Regression, Decision trees, and Neural networks, can generate adequate summaries [12], [13]. The unsupervised tools do not require any training data, so they are suitable for newly observed data without any advanced modifications. Examples of algorithms used here are Clustering and Hidden Markov Model [14][15]. In addition, Semi-supervised algorithms learning techniques require labeled and unlabeled data to generate an appropriate function or classifier.

Compared to extractive methods, abstractive methods are highly complex as they need extensive natural language processing.

2.2.5 Human-Computer Interaction (HCI) approaches

Algorithms of this approach use human actions to create an automatic summary. It is essential to distinguish between actions from the human training process described in the machine learning based approaches, where supervised algorithms use the human work to train the model.

2.2.5.1 Summarization Using Citation

This approach targets scientific papers and scientific areas of research as a textual source. Citations, which can be found in every scientific paper, are written for many reasons. When researchers use an idea or a direct quote from another research, they cite it. The key idea behind these algorithms is that there is enormous information hidden in the citations of scientific papers, and automatically collecting and analyzing this information has many utilizations. A summary is generated using this approach by finding semantic similarity between citations, and the citations with the highest numbers of connections are included in the summary [16][17][18]. Previous works show the importance of citations when trying to know what research is about, even

though each citation of the same article can be very different [19].

2.2.5.2 Summarization Based on Copy Operations

A recent paper proposed a new approach for extractive text summarization: selecting key sentences for summaries based on copy operations of web user [20]. Users copy text from web pages to the clipboard for their purposes [21][22][23]. Users can copy complete sentences for summaries and citations in documentation, blogs, presentations, websites, and answers on forums. It is reasonable to expect that more important sentences would be copied more frequently. Accordingly, by tracking copy operations of web users on a website, summaries can be generated from the most frequently copied sentences.

This thesis extends this approach to text summarization of web pages when data on copy operations of web users is not available.

3 Web Search Solution

3.1 Thesis Goal

The internet is a daily battle for attention. Web users want to find critical information, and they want to find it fast. Most of the users do not read articles till the end [24]. Many do not even scroll down to see information besides the abstract or the lead [25]. Accordingly, short texts can also benefit from summarization. This thesis focuses on a method of automatic text summarization that can be useful also for short texts.

This work has two distinct but related goals. The first goal is to strengthen the indications presented in I. Kirsh and M. Joy paper [20] that sentences that users copy frequently are useful in summaries. The second goal is to create an automatic summarization tool with as few limitations as possible, including regardless of text lengths.

A possible reason for copying information from a web page to the clipboard is to paste it somewhere else on the internet– on forums, web pages, and blogs. Search engines, like Google or Bing, are the primary tools to get information nowadays. If information cannot be found in those search engines, it is almost like it does not exist. A simple example is to think of the last time you wanted to find something in a search engine but failed. Did you blame the engine and search the query elsewhere, or did you think the problem was in your query? Well, I believe the answer is unambiguous. So how exactly can search engines help in text summarization? If copied sentences are often pasted on web pages that search engines index, and sentences copied more frequently are more important - summaries can be generated from sentences that searching them in search engines yield many results. The goal of this thesis was to develop and evaluate a text summarization implementation based on this approach.

3.2 Tools

3.2.1 Wikipedia

As explained above, algorithms for summarizing text with a particular structure and length, like scientific papers, are more common than algorithms for summarizing unstructured data. Moreover, the primary need these days is to summarize data from the internet, as detailed in the introduction section. Those facts led to the decision to use an online platform for the research.

Wikipedia[1], the free encyclopedia anyone can edit, is one of the most famous websites to find information about everything. Wikipedia was launched in January 2001, and today Wikipedia in English has more than 6,300,000 articles available for everyone. Another reason Wikipedia articles can be a good source of text for the research is their

[1] https://www.wikipedia.org/

copyrights policy[2]. The licenses Wikipedia uses grant free access to their content in the same sense that free software is licensed freely. Wikipedia content can be copied and redistributed. This policy is compatible with the hypothesis of this work, as every user can copy text to other websites without special permission. Furthermore, Wikipedia has a dedicated API[3] to approach data from the articles, a limitless web service that allows receiving articles data. I chose Wikipedia as the online source platform and used their top visited articles for the experiments in this work.

3.2.2 Microsoft Bing

Microsoft Bing[4] is a web search engine owned and operated by Microsoft. It is the third-largest search engine globally by market share[5], behind Google and Baidu. Google API strictly limits user's requests per time window, and Baidu is prevalent only in the Chinese market. Therefore, Bing was the natural selection of a search engine for this thesis.

3.3 Web Search Algorithm

The algorithm presented in this thesis is called the WebSearch algorithm, and its stages are described in the following subsections.

3.3.1 Fetching Text and Extracting Sentences

The first step is receiving the original text. The algorithm can receive the text manually, as a URL to the Wikipedia article, or just the article's title as written in Wikipedia. If the algorithm receives the URL or the title, it uses Wikipedia API to fetch the data. The API returns the data as one long string of characters. Then they are carefully separated into sentences (not every dot in a sentence indicates a start of a new sentence). For this purpose, the algorithm uses a regular expression that can identify the end of a sentence with the

[2] https://en.wikipedia.org/wiki/Wikipedia:Copyrights
[3] https://en.wikipedia.org/w/api.php
[4] https://www.bing.com/
[5] https://en.wikipedia.org/wiki/Microsoft_Bing

least false-positive results possible. The text is also cleaned by removing foreign language characters.

3.3.2 Finding the Frequencies of Sentences

The algorithm evaluates the importance of sentences with the help of the Bing search engine. The algorithm queries each sentence in the search engine inside quotation marks. The quotation marks indicate that only precise matches of the complete sentences should be considered. It allows the algorithm to focus on the importance of the complete sentence, eliminating the impact of individual word frequencies.

3.3.3 Generating a summary

Now to return the summary, the algorithm only needs to know the wanted length. The length can be fixed or can be relative to the length of the original text. When it is relative, as in the evaluation process below, the number of sentences in a summary is calculated as the number of sentences in the original article * wanted summary length as a fraction, where the result is rounded down to an integer number.

The algorithm returns the most frequent sentences as an extractive summary of the original text. Figure 1 illustrates the summarization process.

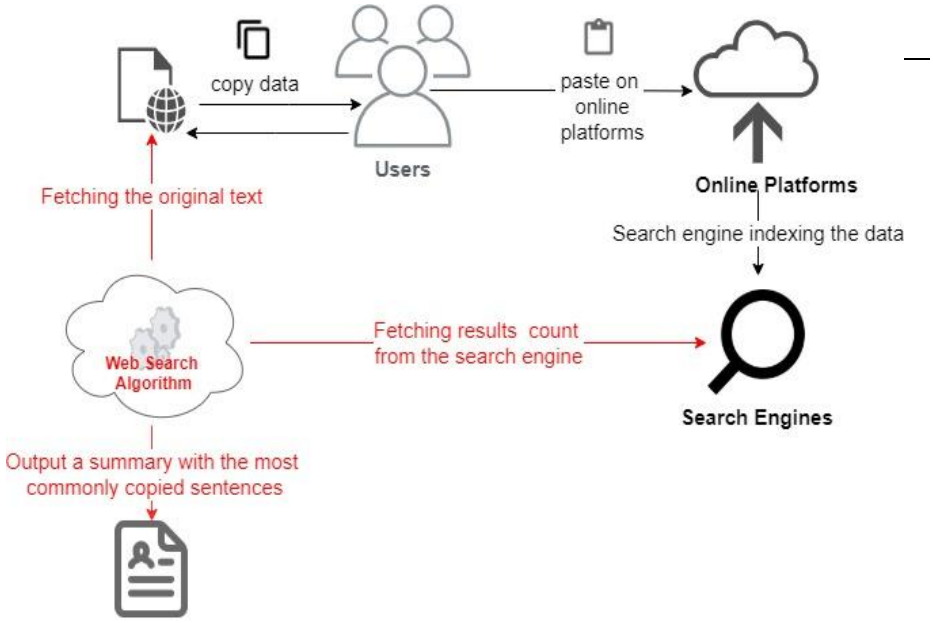

Fig 1. Illustrates the summarization process. Elements in red are parts of the WebSearch stages

4 Evaluation

4.1 Evaluation Process

For the evaluation process, 160 of the most viewed articles in English Wikipedia were chosen[6]. This thesis focuses on a text summarization tool that can select the most important sentences in a short text. To evaluate the algorithm on short texts, I used the Wikipedia API option for fetching only the lead section of each article. The length of the text sources is between 10 to 45 sentences, with an average of 20 sentences. Most of the other summarization tools do not work, by definition, on such short texts.

Evaluating a text summarization tool is not an easy task. The perfect summary of a text is subjective and can be affected by a user's previous knowledge, thoughts, and purpose of the summary. Therefore, there is no one perfect summary, and there is no ground truth.

With that being said, a large enough human survey, where people rank sentences or choose their summary, can be used to evaluate automatic summaries. Similarly, new algorithms can be evaluated using well-known algorithms that have already been found to be effective.

[6] en.wikipedia.org/wiki/Wikipedia:Multiyear_ranking_of_most_viewed_pages

To evaluate the WebSearch algorithm, we can compare its results to an extractive summarization approach algorithm, but which? As discussed in section 2, graph-based algorithms can also create a summary from a small text source, and they do not require previous data on the topic, so they seem to fit the purpose. The two most popular and known graph-based algorithms – LexRank and TextRank, were used. After LexRank and TextRank build a sentence graph, as explained in section 2, they assign a score for each sentence. The higher the score, the more critical the sentence is in the eyes of the algorithm.

The control group in the research was created with a random summary algorithm. The random algorithm ranks the sentences in a completely random way and uses that rank to create a summary. The random algorithm ran twice on each article to create two different random summaries for each article.

The evaluation process measured the similarity between each article's summaries – the WebSearch algorithm's summary, the summary created by LexRank, the summary created by TextRank, and the two random summaries. When comparing only extractive summaries, the evaluation ignores a sentence's position or ranking in the summary while calculating the similarity. If a sentence appears in both summaries, it does not matter where, its similarity rank gets a higher score. Another parameter used in the evaluation process is the length of the summary. As discussed above, summaries can have different lengths. The evaluation process compared the algorithms' results for five different summary lengths – 10%, 20%, 33%, 40%, and 50% of the original text length.

There are several stages in the evaluation process. For every two pairs of algorithms, the evaluation process counts how many sentences in one summary are included on the other algorithm's summary and divides it by the number of sentences in the summary. That calculation returns the

similarity percentage of one article summary between two algorithms and is repeated for each article. The process sums the results and divides them by the number of articles - to receive the average similarity percentage between two summarization algorithms. This process was repeated five times for the five different summary length percentages. Figure 2 illustrates that process in pseudo-code.

```
Algorithms list = {"webSearchCount", "lexRank", "textRank", "random1", "random2"}
Lengths list = {0.1, 0.2, 0.33, 0.4, 0.5}

For each length from lengths list called summaryLength

    For each algorithm from algorithms list called algo1:

        For each algorithm from algorithms list called algo2 != algo1:
        countPercentageOfSimilarities = 0

            For each articles from ObjectDB:
            summary of algo1 = getSummaryOfArticleByAlgoAndLength()
            summary of algo2 = getSummaryOfArticleByAlgoAndLength()
            percentageOfSimilarity = number of sentnces in both summaries / number of sentences in the summary

            countPercentageOfSimilarities + = percentageOfSimilarity

        Average percentage of similarities between algo1 and algo2, with
        summary length of summaryLength = countPercentageOfSimilarities / number of articles


getSummaryOfArticleByAlgoAndLength implementation:

    summaryLength = (int) (sentences.size() * wantedSummaryLengthInFraction);
    sort sentences by wanted algo values
    summary = sentences.subList(0, summaryLength);
```

Fig 2. Illustrates of the evaluation process made by the algorithm to find the percentage of similarities between the algorithms

4.2 Evaluation Results

Five similarity tables have been generated. Each table presents ten different percentages of similarity between algorithm pairs. Important note: The percentage shown does not relate to the number of similar words on the summary. We are comparing extractive summaries, so each sentence is either in or out of the summary. All sentences are considered equal regardless of their length.

Table 1. Average Percentage of Similarity with Summary Length of 50% of the Text

	Text Rank	Lex Rank	Random 1	Random 2
Web Search	84.91%	90.77%	49.53%	51.21%
Text Rank	-	75.44%	51.50%	50.53%
Lex Rank	-	-	48.87%	51.33%
Random 1	-	-	-	49.08%

In Table 1, we can see the first promising results. Here the average summary length is ten sentences. Although a summary builds from 50% of the original text is not so common or valuable, and this comparison made mainly to cover extreme use cases, the significance of the similarity between the Web Search algorithm to the TextRank (84.91%) and LexRank (90.77%) summaries worth noting.

Another diagnosis from Table 1 is related to the similarity of TextRank and LexRank (75.44%). This result is vital to the logical consequence process. This relatively low percentage may confirm that although TextRank and LexRank algorithms are of the same graph-based approach, they are different enough to indicate that the evaluation process uses two different algorithms, not just one with a few minor

changes. In addition, the random algorithm results serve as control groups. Their similarity percentage remains close to the summary length in percentage, as one would expect to see.

Table 2. Average Percentage of Similarity with Summary Length of 40% of the Text

	Text Rank	Lex Rank	Random 1	Random 2
Web Search	81.52%	88.75%	39.01%	37.78%
Text Rank	-	70.15%	38.94%	38.23%
Lex Rank	-	-	39.10%	38.77%
Random 1	-	-	-	38.68%

Table 2 shows similar results to Table 1. Here the average summary length is eight sentences. A 40% summary length is still not a very common summary length, but it is positive to see that even though the summaries lengths changed by a significant percentage, the direction of the results remains the same.

Table 3. Average Percentage of Similarity with Summary Length of 33% of the Text

	Text Rank	Lex Rank	Random 1	Random 2
Web Search	78.62%	87.27%	30.97%	30.00%
Text Rank	-	65.72%	30.25%	30.65%
Lex Rank	-	-	29.57%	31.12%
Random 1	-	-	-	29.81%

Table 3 is the most important, in my opinion. Here the average summary length is six sentences. The similarity between the algorithm results to a known and proven algorithm like LexRank may show that using the "crowd wisdom" may have a place in the future automatic summaries research world.

Table 4. Average Percentage of Similarity with Summary Length of 20% of the Text

	Text Rank	Lex Rank	Random 1	Random 2
Web Search	77.33%	84.35%	17.38%	18.06%
Text Rank	-	61.55%	18.26%	17.44%
Lex Rank	-	-	16.70%	19.44%
Random 1	-	-	-	19.74%

Table 4 strengthens the thoughts about the relatively significant differences in LexRank and TextRank. Here the average summary length is four sentences. On average, almost 40% of their summaries are different when creating a summary from 20% of the text sentences. As mentioned, that diagnosis is essential to the logical consequence process.

Table 5. Average Percentage of Similarity with Summary Length of 10% of the Text

	Text Rank	Lex Rank	Random 1	Random 2
Web Search	72.97%	81.51%	5.94%	6.41%
Text Rank	-	51.67%	8.13%	5.99%
Lex Rank	-	-	4.79%	6.41%
Random 1	-	-	-	11.77%

Table 5 is the last table of results from the evaluation process, and here, the average summary length is just two sentences, the two most important sentences in a text. Two diagnoses can be inferred. The first one is about the control group. We can see that the three non-random methods have relatively high similarity percentages. At the same time, comparison with the random algorithm shows low similarity percentages, as expected. The second diagnosis is that, again, the high similarity between the WebSearch algorithm and the graph-based algorithms is especially impressive when considering that the extractive summaries are built from only 10% of the sentences of the original text.

We can see that the similarity percentages when the random algorithm is included are close to the summary lengths in percentages but tend to be a bit lower. This is probably because the summary length was rounded down to an integer number (the summaries include only complete sentences), so the actual summaries were shorter than the length parameters. The effect is especially noticed in Table 5, as the impact of rounding down short summaries is larger.

The comparison of extractive summaries is made by only one method, a similarity percentage, while abstractive summaries can be evaluated with different comparison tools. Hence, presenting the results in another view could shed more light here. As discussed above, there is no perfect summary, and therefore, there is no absolute ground truth. However, for evaluation purposes, let us assume that LexRank and TextRank, well-known effective algorithms, represent the ground truth. In table 6, the assumption is that the summaries created by LexRank are the ground truth, and the other algorithms are evaluated according to their similarity to LexRank. Table 7 is similar, with TextRank as the evaluator. The highest similarity results for each summary length are marked in bold.

Table 6. Average Percentage of Accuracy when LexRank is Considered as Ground Truth

Algorithms / Lengths	Web Search	TextRank	Random1	Random2
10%	**81.51%**	51.67%	4.79%	6.41%
20%	**84.35%**	61.55%	16.70%	19.44%
33%	**87.27%**	65.72%	29.57%	31.12%
40%	**88.75%**	70.15%	39.10%	38.77%
50%	**90.77%**	75.44%	48.87%	51.33%

Table 7. Average Percentage of Accuracy when TextRank is Considered as Ground Truth

Algorithms / Lengths	Web Search	LexRank	Random1	Random2
10%	**72.97%**	51.67%	8.13%	5.99%
20%	**77.33%**	61.55%	18.26%	17.44%
33%	**78.62%**	65.72%	30.25%	30.65%
40%	**81.52%**	70.15%	38.94%	38.23%
50%	**84.91%**	75.44%	51.50%	50.53%

Of course, neither LexRank nor TextRank produces perfect summaries. However, tables 6 and 7 present interesting and encouraging evaluation results. These results indicate a correlation between the frequency of sentences in web search to their potential in text summaries. The correlation is strengthened when discerning that the WebSearch summaries received the best evaluation by LexRank and TextRank in all the tested summary lengths.

4.4 Examples of Summaries

The comparison tables above show a macro view of the results, but an example could shed light on the differences between the algorithms. The example below examines the World War I article. The article's lead contains 39 sentences, and the chosen summary length is 33% or 12 sentences (after rounding down). The LexRank and WebSearch algorithms output the following summaries:

Table 7. LexRank Algorithm Summary

Sentence	score
Serbia's reply failed to satisfy the Austrians, and the two moved to a war footing.	1.8083
Contemporaneously known as the Great War or the war to end all wars, it led to the mobilisation of more than 70 million military personnel	1.3350
The Triple Alliance was only defensive in nature, allowing Italy to stay out of the war until April 1915, when it joined the Allied Powers	1.2995
World War I was a significant turning point in the political, cultural, economic, and social climate of the world.	1.2974
In 1915, Italy joined the Allied Powers and opened a front in the Alps.	1.2211
The war and its immediate aftermath sparked numerous revolutions and uprisings.	1.1975
Russia felt it necessary to back Serbia, and approved partial mobilisation after Austria-Hungary shelled the Serbian capital of Belgrade	1.1975
On 2 August, Germany demanded free passage through Belgium, an essential element in achieving a quick victory over France.	1.1766
The Big Four (Britain, France, the United States, and Italy) imposed their terms on the defeated powers in a series of treaties agreed	1.1766
Though Serbia was defeated in 1915, and Romania joined the Allied Powers in 1916, only to be defeated in 1917, none of the great powers were knocked out of the war until 1918	1.1236
Ultimately, as a result of the war, the Austro-Hungarian, German, Ottoman, and Russian Empires ceased to exist	1.1236
In the end, World War I would see the continent of Europe split into two major opposing alliances	1.0984

Table 8. WebSearch Algorithm Summary

Sentence	Web Results
Serbia's reply failed to satisfy the Austrians, and the two moved to a war footing.	1440
World War I or the First World War, often abbreviated as WWI or WW1, was a global war originating in Europe	1380
A network of interlocking alliances enlarged the crisis from a bilateral issue in the Balkans to one involving most of Europe.	861
Contemporaneously known as the Great War or the war to end all wars, it led to the mobilisation of more than 70 million military personnel	764
By July 1914, the great powers of Europe were divided into two coalitions: the Triple Entente, consisting of France, Russia, and Britain; and the preestablished Triple Alliance of Germany, Austria-Hungary, and Italy.	87
The Triple Alliance was only defensive in nature, allowing Italy to stay out of the war until April 1915, when it joined the Allied Powers	80
World War I was a significant turning point in the political, cultural, economic, and social climate of the world.	80
The German advance into France was halted at the Battle of the Marne and by the end of 1914	76
The war and its immediate aftermath sparked numerous revolutions and uprisings.	76
On 2 August, Germany demanded free passage through Belgium, an essential element in achieving a quick victory over France.	75
The Big Four (Britain, France, the United States, and Italy) imposed their terms on the defeated powers	74
Though Serbia was defeated in 1915, and Romania joined the Allied Powers in 1916, only to be defeated in 1917, none of the great powers	73

Table 9. Sentences in WebSearch Algorithm Summary but not in LexRank Algorithm Summary

World War I or the First World War, often abbreviated as WWI or WW1, was a global war originating in Europe
A network of interlocking alliances enlarged the crisis from a bilateral issue in the Balkans to one involving most of Europe.
By July 1914, the great powers of Europe were divided into two coalitions: the Triple Entente, consisting of France, Russia, and Britain; and the preestablished Triple Alliance of Germany, Austria-Hungary, and Italy
The German advance into France was halted at the Battle of the Marne and by the end of 1914

Table 10. Sentences in LexRank Algorithm Summary but not in WebSearch Algorithm Summary

In 1915, Italy joined the Allied Powers and opened a front in the Alps
Russia felt it necessary to back Serbia, and approved partial mobilisation after Austria-Hungary shelled the Serbian capital of Belgrade
Ultimately, as a result of the war, the Austro-Hungarian, German, Ottoman, and Russian Empires ceased to exist
In the end, World War I would see the continent of Europe split into two major opposing alliances

Looking at two summaries of one text source cannot draw clear conclusions, but interesting noting can be found when examining the differences. The original article's first sentence – " World War I or the First World War, often abbreviated as WWI or WW1, was a global war originating in Europe", is in the WebSearch algorithm and not in LexRank algorithm. We can assume that web users copy this sentence on many online platforms when discussing World War I. On the other hand ,it would be reasonable to say that the LexRank algorithm did not include this sentence because most of its content is a synonym for the name of the war and does not appear again in the text. To declare whether this sentence should be in summary is subjective, but it could indicate a possible advantage of the WebSearch algorithm and the HCI summary approach over the LexRank algorithm.

5 Conclusions

This thesis presents a novel approach to extractive text summarization: using the frequency of sentences in web search engines to evaluate the importance of sentences. The World Wide Web today contains virtually unlimited data and information. The internet today is the source of information and the place to share it. When people find a meaningful and relevant piece of information, they often reuse it – creating summaries, using it as citations in documentation, answering

on forums, writing it in their blogs, or presenting it on their presentations or websites.

In this thesis, I continued to explore the idea of I. Kirsh and M. Joy[20] that data regarding what users copy on websites can help estimate the importance of sentences. According to their idea, I created a summarization tool of textual data that tries to select the most crucial information in a text with the help of a web search engine.

I evaluated the summaries by comparison to popular and proven extractive text summarization algorithms. The results indicate that the two goals of this thesis have been achieved. First, the assumption about "crowd wisdom" importance has strengthened. The results show that data copied and used more frequently are likely to be included in a summary. Second, the extractive summarization algorithm created by the "crowd wisdom" assumption could be helpful in conditions where other methods could not work. It has the advantages mentioned - it does not depend on tagged data and natural language processing calculations, it does not assume having the text topic, and it can find the most critical information even when getting a minimal amount of textual data.

With that being said, in the process, I found two disadvantages of the algorithm. The first disadvantage is that although it does not require any pre-processing or training process, the algorithms running time might take longer than other algorithms. Fetching result counts from the search engine takes a few milliseconds, but that process might take a long time for a significant source of textual information. The second disadvantage is more restricting, in my opinion. The algorithm search results for each sentence inside quotation marks, so it cannot work on a new text source. As explained in the HCI Solution section, querying with quotation marks is essential. Without quotation marks, the algorithm rates a sentence according to the frequency of each word in it, which can cause erroneous results. Because the

"training" of the algorithm, meaning the process of users copying and pasting the text, does not work by demand but rather happens naturally over time, new or modified sentences may have a small number of results in search engines, so this has to be taken into consideration.

6 Future Work

I believe the research area of automatic summarization tools in general, and for small text sources in particular, has vast growth potential. The demand for reliable, consistent, fast, and fluent automatic summarization will grow with the growth of information on the internet, which the forecasts predict to be tremendous. I think the results of this thesis should be considered when trying to find a better solution to the automatic summarization problem than the currently available solutions to summarize relatively small text sources. As mentioned before, it is hard to find the ground truth to evaluate automatic summarization tools, but I believe that future research involving manual human evaluation can prove the correlation between the importance of a sentence and its frequency on a web search. These results may set the ground for using the "crowd wisdom" approach in future summarizations tools.

6. Bibliography

[1] M. Gambhir and V. Gupta, “Recent automatic text summarization techniques: a survey,” *Artif. Intell. Rev. 2016 471*, vol. 47, no. 1, pp. 1–66, Mar. 2016, doi: 10.1007/S10462-016-9475-9.

[2] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining Text Data*, vol. 9781461432, Springer US, 2012, pp. 43–76.

[3] A. P. Widyassari *et al.*, “Review of automatic text summarization techniques & methods,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2020, doi: 10.1016/J.JKSUCI.2020.05.006.

[4] A. Helen, “Automatic Abstractive Summarization Task for News Article,” *Emit. Int. J. Eng. Technol.*, vol. 6, no. 1, pp. 22–34, Jul. 2018, doi: 10.24003/EMITTER.V6I1.212.

[5] H. Saggion and T. Poibeau, “Automatic Text Summarization: Past, Present and Future,” pp. 3–21, 2013, doi: 10.1007/978-3-642-28569-1_1.

[6] U. R. Udo Hahn, “Knowledge-based text summarization: Salience and generalization operators for knowledge base abstraction,” *MIT Press. Cambridge, Mass*, pp. 215–232, 1999.

[7] P. Bouquet, C. Ghidini, F. Giunchiglia, and E. Blanzieri, “Theories and uses of context in knowledge representation and reasoning,” *J. Pragmat.*, vol. 35, no. 3, pp. 455–484, Mar. 2003, doi: 10.1016/S0378-2166(02)00145-5.

[8] S. Harabagiu and F. Lacatusu, “Topic Themes for Multi-Document Summarization,” in *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil*, 2005, pp. 202–209, Accessed: Sep. 07, 2021. [Online]. Available: http://duc.nist.gov.

[9] G. Erkan and D. R. Radev, “LexRank: Graph-based

lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, 2004, doi: 10.1613/jair.1523.

[10] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text." pp. 404–411, 2004, Accessed: Sep. 07, 2021. [Online]. Available: https://aclanthology.org/W04-3252.

[11] L. Page and S. Brin, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Networks ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998, doi: 10.1016/S0169-7552(98)00110-X.

[12] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," *Comput. Speech Lang.*, vol. 23, no. 1, pp. 126–144, Jan. 2009, doi: 10.1016/J.CSL.2008.04.002.

[13] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," *Appl. Intell. 2013 404*, vol. 40, no. 4, pp. 592–600, Dec. 2013, doi: 10.1007/S10489-013-0490-0.

[14] L. Yang, X. Cai, Y. Zhang, and P. Shi, "Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization," *Inf. Sci. (Ny).*, vol. 260, pp. 37–50, Mar. 2014, doi: 10.1016/J.INS.2013.11.026.

[15] J. Conroy and D. P. O'leary, "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition," 2001.

[16] S. Teufel, A. Siddharthan, and D. Tidhar, "Automatic classification of citation function," *Proc. 2006 Conf. Empir. Methods Nat. Lang. Process.*, pp. 103–110, 2006, doi: 10.5555/1610075.1610091.

[17] E. Collins, I. Augenstein, and S. Riedel, "A Supervised Approach to Extractive Summarisation of Scientific Papers," *CoNLL 2017 - 21st Conf. Comput. Nat. Lang. Learn. Proc.*, pp. 195–205, Jun. 2017, Accessed: Sep. 08, 2021. [Online]. Available:

https://arxiv.org/abs/1706.03946v1.

[18] V. Qazvinian and D. Radev, “Scientific Paper Summarization Using Citation Summary Networks.” Manchester, pp. 689–696, 2008, Accessed: Sep. 08, 2021. [Online]. Available: https://aclanthology.org/C08-1087.

[19] A. Elkiss, S. Shen, A. Fader, G. Erkan, D. States, and D. Radev, “Blind men and elephants: What do citation summaries tell us about a research article?,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 1, pp. 51–62, Jan. 2008, doi: 10.1002/ASI.20707.

[20] Ilan Kirsh and Mike Joy, “An HCI Approach to Extractive Text Summarization: Selecting Key Sentences Based on User Copy Operations,” in *In Proceedings of the 22nd HCI International Conference (HCII 2020), Communications in Computer and Information Science. Springer International Publishing, Cham*, 2020, pp. 335–341, doi: 10.1007/978-3-030-60700-5_43.

[21] I. Kirsh, “What Web Users Copy to the Clipboard on a Website: A Case Study,” in *In Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020). INSTICC, SciTePress, Setúbal, Portugal*, 2020, pp. 303–312, doi: https://doi.org/10.5220/0010113203030312.

[22] I. Kirsh, “Automatic Complex Word Identification Using Implicit Feedback From User Copy Operations,” in *In Proceedings of the 21st International Conference on Web Information Systems Engineering (WISE 2020), Lecture Notes in Computer Science. Springer International Publishing, Cham*, 2020, pp. 155–166, doi: https://doi.org/10.1007/978-3-030-62008-0_11.

[23] I. Kirsh, “Word-Copying on a Website as a Word Complexity Indicator and the Relation to Web Users’ Preferred Languages,” in *In Asian CHI Symposium 2021 (Asian CHI Symposium 2021), May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA*,

2021, pp. 16–20, doi: https://doi.org/10.1145/3429360.3468172.

[24] Farhad Manjoo, “You Won’t Finish This Article,” *Slate*, Jun. 2013. https://immagic.com/eLibrary/ARCHIVES/GENERAL/GENPRESS/S130606M.pdf (accessed Sep. 22, 2021).

[25] M. Sidoff, “How People Read Short Articles,” *CXL*, Jan. 2018. https://cxl.com/research-study/people-read-short-articles-original-research/ (accessed Sep. 22, 2021).