Intelligent Text Summarization: a strategy to reduce information misrepresentation

Shreya Jacob, B.Tech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Data Science)

Supervisor: Prof. Khurshid Ahmad

August 2021

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Shreya Jacob

August 31, 2021

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Shreya Jacob

August 31, 2021

Intelligent Text Summarization: a strategy to reduce information misrepresentation

Shreya Jacob, Master of Science in Computer Science

University of Dublin, Trinity College, 2021

Supervisor: Prof. Khurshid Ahmad

The year 2019 would remain in history forever due to the outbreak of Covid-19. Even though it has been two years after the discovery of coronavirus, the situation in many nations remains uncontrollable. According to a few experts, one of the causes is public obliviousness due to distortion of the truth. In an unfamiliar environment, the requirement for precise information is paramount. The information gets diffused on a text cline from scientific papers to science magazine articles, then to newspapers, and then to the general public via social media, where half of it gets lost or corrupted. The accurate findings of the researchers get suppressed. This project work designs an automatic text summarization system based on the theory of lexical cohesion that can efficiently extract the pertinent information from the research papers to reduce the misrepresentation of text in the first level of text cline. The notion of lexical cohesion is that the repetition of words in the sentences creates a bond between them and brings the text closer. Identifying the highly bonded sentences would thus aid in creating a summary that is concise and meaningful. The concept of using an external keyword list that consists of top terms present in the domain for keyword identification was a significant contribution of this work. The system efficiency was statistically evaluated using various metrics like average sentence length, readability, sentiment similarity, and syntactic similarity. The evaluation results of the summaries generated for ten research papers confirmed the efficiency of the algorithm when compared to their abstracts (human-generated summaries). Although the summaries were less readable than the abstracts, they were highly similar to the original text on sentiment and syntactic similarity.

Acknowledgments

I take this opportunity to thank everyone who helped and supported me during this dissertation work.

First and foremost, I would like to thank my supervisor, Prof. Khurshid Ahmad, for offering invaluable advice, consistent feedback, and his patience in explaining topics to me for the entire duration of this dissertation work.

Next, I thank my beloved parents, Litty Jacob and Jacob Mathews, without whom I wouldn't have been writing this at the moment. Their words of encouragement have always been a motivation for me. I also take this opportunity to thank my sister, Sandra, and my brother-in-law, John, for their constant support and motivation during the hard times.

I also thank all my friends and family for their prayers and blessings. Special mention to Bharath for his words of encouragement and our report writing sessions, and to John for proofreading this report and providing valuable suggestions.

Shreya Jacob

University of Dublin, Trinity College
August 2021

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

As the world is fighting the pandemic (COVID-19 outbreak) situation, another alarming issue that has awoken is the lack of authentic information. The existence of fake news is not something that has emerged all of a sudden. Right from the beginning, when the public information started to broadcast, so did the fabrication process. The Oxford dictionary has recently added the term *fake news* and defined it as "the news that conveys or incorporates false, fabricated, or deliberately misleading information or that is characterized as, or accused of doing so." The underlying force behind this is either withholding the truth for the benefit of the powerful or duping the public into believing whatever the media wishes. The rapid evolution of social media has increased the spread of misinformation. Any human can effortlessly spread a piece of information with a single click sitting in their home with no or less verification. This easiness has created a confounding situation for the readers whether to believe everything on social media.

The writing style of a text differs based on the target audience. The scientific papers have a target audience of people with scientific knowledge on the subject. The articles in science magazines are comparatively less complex but intended for an educated crowd, whereas the news articles are solely for the general public. And on the last level, social media posts have minimal difficulty forcing ordinary people to depend on these sources. This difference in writing style creates room for misrepresentation of the information. As Emily has stated in her thesis (Harte (2021)), the distribution of knowledge operates on a text decline or the text-cline, and Figure 1.1 captures the downward flow of text-cline.

While fake news has always been a concern in our daily life, the need for accurate information is at most during an unfamiliar environment. When the outbreak of Covid-19 was confirmed, even the scientists who discovered it knew little about it. But, in haste to

publicize any new information that they receive, even with less credibility for sensationalism, the media has ended up manipulating the truth. And the power of social media has further amplified this situation. The findings of the researchers have been twisted up and delivered differently. The consequences of it are non-negligible as conditions like that of pandemic affect the health of all personalities. Although we have survived two years past the coronavirus breakdown, most people are not aware of the accurate information regarding the virus or that of the vaccines. And one of the reasons for not eradicating the virus is this lack of awareness among the people.

Figure 1.1: A Visual representation of text cline (Harte (2021))

1.2 Problem definition

As discussed, the consequences of fake news during a pandemic are non-negligible. There have been numerous researches that distinguished the text as spurious or genuine. However, accurate identification of fake news is still challenging due to the dynamic nature of social media and the complexity and diversity of online communication data (Zhang and Ghorbani (2019)). There are various fact-checking tools available online that verify the credibility of the given text using machine learning approaches or even manual verification from community users like the ones initiated by Facebook (Mosseri (2019)). The problem with such methods is that the result is either true or false and assigning such values is difficult as the content might be a mixture of it. Moreover, such machine learning approaches require a huge well-labeled dataset for the training process. This project, therefore, aims to provide a solution for the fake news spread taking into consideration

the COVID-19 scenario.

The authors of science magazines refer to research journals for the content. The lengthy scientific papers are tiresome to read that they often resort to the abstract for the content. The abstracts are for experts by experts, and sometimes only for experts that the language is terse and requires substantial background knowledge (Benbrahim and Ahmad (1995)). The project thus aims to automatically summarize the scientific papers without omitting any pertinent information and use them as guidelines for the articles and reduce the misrepresentation on the first level. If this is achievable, it would help to reduce the distortion in the further stages. The outcome of this approach would also be beneficial for the scientific papers without an abstract.

Automatic text summarization creates a short paragraph that conveys the same meaning as the original text making it easier for the reader. In the current world, where we are flooded with millions of textual data each day, it has become necessary to have a system that can summarize long text to abridged versions. For the same reason, many researchers have proposed various methods since 1958 (Luhn (1958)). Extractive and abstractive are two types of text summarizing processes based on the summary produced. The extractive method finds the most significant sentences from the text and ranks them to create an abridged version. It means the result would have the sentences that are in the original text. The abstractive summarization forms new sentences identifying the overall message of the original text to produce the output. The abstractive method is more complex and problematic since it demands the formation of sentences not in the text, which may alter the original meaning, and the evaluation of such a summary necessitates manual tasks. The scientific papers are usually elaborate with repetition of the same idea. Hence, it is sufficient to implement the extractive text summarization method to create a quality summary of research papers.

The key idea of extractive text summarization is the identification of the relevant sentences in minimal time. Most of the existing methods use machine learning approaches to identify the key sentences that require a large dataset and training time. Another disadvantage of these methods is that they are specific to the training data's topic domain, requiring the model to be trained on a dataset with the same field. As a result, there is a need for a simple system that can summarize the text in a specific domain without requiring extensive training or enormous datasets.

1.3 Contributions

The dissertation aims to stop fake news from spreading during a pandemic by summarizing scientific publications and conserving critical information. This work is an extension of

the thesis by Harte (2021). An extractive summarization system is created based on the theory of lexical cohesion by Hoey (1991). It follows the idea that repetition of words in different sentences would create a bond between them and that identifying sentences with strong bonds will aid in creating a good summary.

The summarization algorithm identifies the keywords in the text, ignoring all the close class words, and looks for simple repetition, complex repetition, simple paraphrase, or complex paraphrase of the keywords between each sentence. The system enables using an external keyword list of the most common words discovered in COVID-19-related texts or all of the text's nouns as keywords. The algorithm created can be reused for any other subject domain by changing the keyword list. One of the project's significant contributions is identifying the missing named entities in the keyword list, such as vaccine-producing firms or the names of notable coronavirus researchers.

The project also contributes to the automated evaluation of the summaries generated. The summary is statistically evaluated based on readability measures, lexical similarity, syntactic similarity, and sentiment similarity. It would present the summary's efficacy by comparing the readability, word distribution, meaning, and sentiment level to the abstract of the text (human-generated summary). Popular magazines in science such as *Science* and *The Scientist* are scraped into a dataset composed of ten articles and referenced research papers. Both the article and research paper are summarized and statistically evaluated.

1.4 Structure of dissertation

The following is how the rest of the dissertation is structured. An overview of related research on extractive text summarization, lexical cohesion, and evaluation techniques is presented in the next chapter(See chapter 2). It is followed by the methodology used for the summarization algorithm and the method for summary evaluation (See chapter 3). The data collection and processing, the experiments implemented to prove the efficiency of the proposed method, and the results of the summary evaluation are detailed next (See chapter 4). The final chapter consists of the limitations of the proposed method, conclusion, and future work suggestions (See chapter 5).

Chapter 2

State of the Art

The previous chapter described the project's objective, motivation, and contributions. This chapter presents the background of the topic and details the methods used and the related works on text summarization and evaluation techniques.

2.1 Background

Fake news or the withholding of truth has always been an alarming concern around the world. Its impact intensifies in instances where the entire crowd is clueless. Such a scenario happened when the world got hit by the coronavirus outbreak. The ease of use of social media has created a massive platform for the spread of misleading information. The BBC news investigated the fake news spread during the pandemic to identify the types of people behind it (Spring (2020)). They found that the misleaders include pranksters, politicians, conspiracy theorists, insiders from trustworthy sources, celebrities, or even ordinary people to save their friends and family just in case the information turned out to be true.

Many researchers have been researching different methods to solve the issue of fake news. The identification of fake news is a daunting task. First, classifying a text as true or false is ambiguous as the content would be a mixture. Secondly, the fake news spread out today is usually through social media without a reference to the source. Thus comparing the news post to the source is not applicable anymore. The advancement of various deep learning algorithms such as the Recurrent Neural Network and Auto Encoder has aided researchers in combating the social media fake news spreading (Zhang and Ghorbani (2019)). Even yet, the method is not easy because supervised learning necessitates a massive labeled dataset. There has been very little research for unsupervised learning methods for fake news detection. The idea of summarizing the scientific papers to re-

duce the misrepresentation of text in the first level of text cline thus helps to reduce the origination of fake news in subsequent levels of text-cline. Automatic text summarization is a well-researched topic due to its various use cases. Saseedran (2019) has highlighted some of the areas where automated text summarization is applicable. It includes automated content creation, book summarization, e-learning and class assignments, financial research, helping disabled people, and many more. This project uses text summarization to combat infodemics during a pandemic.

Automatic text summarization can be classified differently based on purpose, input, or output (Chauhan (2018)). It is categorized into three types based on its purpose: generic, domain-specific, and query-based. The context of the text is not taken into account in generic summarizing. This method's model can be used to summarize any subject content. In domain-specific summarization, the model is built around a single domain. As a result, only texts from that domain can be summarized. The summary in query-based is generated depending on the terms in the query. Based on the input text, there are two summarization techniques: single-document and multi-document. In single-document summarization, the input text is short and consists of only one document whereas, in multi-document, the input text may consist of more than one document and is usually very long. Based on the output type, there is extractive and abstractive summarization. Extractive summarization picks the most important sentences from the input text and creates the summary. Abstractive summarization transforms the sentences from the input and generates new ones. Figure 2.1 from (Chauhan (2018)) depicts the same. In this project, an extractive summary method with a single document input is constructed, with the option of applying domain-specific or generic information during summarizing.

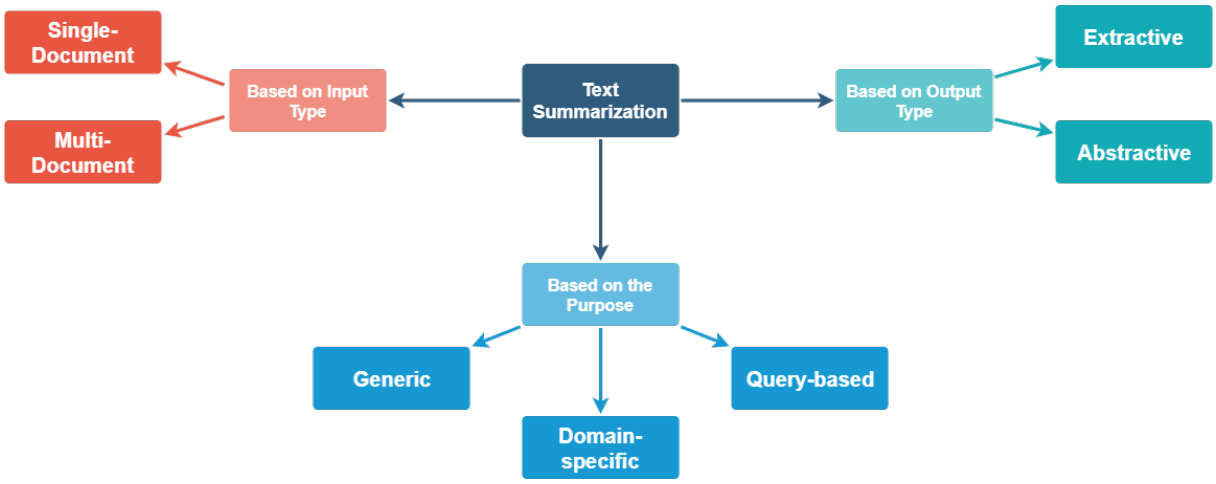

Figure 2.1: Classification of text summarization (Chauhan (2018))

As Ercan and Cicekli (2008) explains, a meaningful text has semantic integrity to explain a topic and is not a random sequence of words. This semantic integrity is termed *coherence* in linguistics. Coherence is defined as "continuity of senses" and "the mutual access and relevance within a configuration of concepts and relations" by De Beaugrande and Dressler (1981). It is the internal element that establishes a theme and brings the logic in the text to life. The external factor that brings the text closer, on the other hand, is *cohesion*. Halliday et al. (1976) defines it as "the way of getting text to hang together as a whole." Cohesion is much simpler than coherence and deals directly with the relationship between text units (Ercan and Cicekli (2008)). Cohesion in a text can be syntactic or lexical. Syntactical cohesion is language-based and expresses cohesion between the words in a closed class. Lexical cohesion is the repetition of the words in a text that creates a tie or bond between the sentences. Hoey (1991), in his *Patterns of Lexis in Text* elaborates how the repetition of words in a text brings cohesion. Following this idea, Benbrahim and Ahmad (1995) created TELE-PATTAN, a cohesion-based summarization system. A modified version of their work is carried out in this project.

The authors of scientific papers write them to convince the reader of a new method discovered by them. When the writer has to convince something, they end up using the exact words or their synonyms repeatedly. This concept presents an ideal position for summarizing scholarly papers using the lexical cohesion technique.

2.2 Theory of Lexical Cohesion

According to Halliday et al. (1976), a mere occurrence of cohesiveness or cohesively linked items is considered as a *tie* in a text. They have classified the cohesive tie into five classes: reference, substitution, ellipses, conjunction, and lexical cohesion. Table 2.1 shows the taxonomy of cohesive ties by Halliday et al. (1976). The first four classes fall under grammatical or syntactic cohesion. *Reference* occurs in a text when a sentence refers to an item previously introduced by another sentence. It can be personal (using pronouns), demonstrative (using determiners), or comparative (Nawaz (2014)). *Substitution* and *Ellipses* both replace a clause without repeating it in its next occurrence. The difference between them is that substitution would replace the clause with its grammatical synonym whereas, ellipses replace it without any additional word. *Conjunction* occurs when the text uses conjunctions (and, but, etc.) to create links between the sentences. *Lexical cohesion* is based on the lexical item (words) rather than the meaning of the text. It is further categorized as reiteration and collocation. Reiteration tie occurs when there is a repetition of the exact word, plural/singular form, synonym, or super-ordinate form. Collocation tie is the term for the words with a certain probability of occurring together

Table 2.1: Classification of cohesion based on Halliday et al. (1976)

Grammatical Cohesion	**Reference** - Sentence refers to an item previously introduced	**Personal** - Using pronouns, eg. Eve was walking in the Garden of Eden when the serpent slithered over to her. **Demonstrative** - Using determiners, eg. Jim had one loaf of bread. He gave the bread to Alice. **Comparative** - performing comparison, eg. Eve need to look more beautiful than other women
	Substitution - Replaces one clause by another	eg. "You need to look more beautiful for your man". "No, I don't"
	Ellipsis - Replaces a clause by nothing	eg. How many potatoes do you want, Sir? – Four [], please. Here potatoes are replaced by none.
	Conjunction - linkage between sentences using conjunctive words	**Additive** - adding two clauses, eg. A Visitor Arrives from Morocco and tells me a curious story. **Adversative** - expressing opposition, eg. He has no other women but me.
Lexical Cohesion	**Reiteration** - Repetition of same word	eg. Raechel has a dog named Simba. Simba is very loyal to her.
	Collocation - Words occuring together	eg. It started to rain heavily that Sarah had to take out her umbrella.

in a text. The words 'man' and 'woman' identifies as collocated as they are opposite. But the words like 'rain' and 'umbrella' are also collocated terms though they don't convey the same meaning. This relationship between the words is difficult to model as they don't depend on any general semantic relationship but rather belong to the same context that they appear together (Cerban (2010)). Hasan and Flood (1984), in their later work, replaced the categories of lexical cohesion as shown in Table 2.2 to include some sub-categories of the collocation.

Hoey (1991) has further detailed the idea of lexical cohesion in text. According to him, two sentences with a repetition share a *link*. The link here is the same as *tie* by Halliday et al. (1976). Repetition is classified into four by Hoey (1991): simple repetition, complex repetition, simple paraphrase, complex paraphrase. Simple repetition occurs when the exact word repeats or the singular/plural version repeats. For example, 'vaccines' in a sentence would be a simple repetition of 'vaccine'. Complex repetition occurs when two words with the same lexical morpheme and without the same grammatical meaning exist. For example, 'drug' and 'drugging' shares the same morpheme 'drug' but has different

Table 2.2: The sub-categories of lexical cohesion according to Hasan and Flood (1984)

General	**Repetition**	leave, leaving, left
	Synonymy	leave, depart
	Antonymy	leave, arrive
	Hyponymy	travel, leave (including co-hyponyms, leave, arrive)
	Meronymy	hand, finger (including co-meronyms, finger, thumb)
Instantial	**Equivalence**	the sailor was their daddy; you be the patient, I'll be the doctor
	Naming	the dog was called Toto; they names the dog Fluffy
	Semblance	the deck was like a pool; all my pleasures are like yesterdays

grammatical meaning. Simple paraphrase occurs when a synonym (conveying the same meaning) of a word repeats. For example, 'sedated' and 'tranquilized' convey the same idea. Complex paraphrase occurs in two conditions. In the first case, there is a repetition of antonym (hot-cold). The second scenario occurs as a result of a link. A complex repetition of A(B) and a simple paraphrase of A(C) would make a complex paraphrase between B and C. For example, 'writer' and 'writings' have a complex repetition relationship, and 'writer' and 'author' have a simple paraphrase relationship (synonym). It would make 'author' and 'writings' have a complex paraphrase relationship. The link between A and C can also be antonym.

Any two sentences with the repetitions mentioned above form a link. Hoey (1991) then establishes that any two sentences with more than three links to be considered as sharing a bond. Further, he categorizes each sentence as topic opening, topic closing, central, and marginal. Marginal sentences are those that have no bonds or few bonds. Central sentences are those that are highly bonded, "The most bonded sentences". The sentences with bonds more than the bond strength are classified as topic-opening if they have above-average bonds with succeeding sentences than preceding sentences and topic-closing for vice versa. Here, the bond strength is depended on the text but is usually fixed as 3. Avoiding the marginal sentences and selecting the most-bonded sentences from other classes can aid in creating the summary.

2.3 Related Works

According to (Widyassari et al. (2020)), the most common method for document summarization is fuzzy logic, Term Frequency-Inverse Document Frequency (TF-IDF), and Latent Semantic Analysis (LSA). The Fuzzy Logic method evaluates the different characteristics of sentences such as Frequency, Similarity, Position, and Length where each

feature can be weighted differently. The TF-IDF approach identifies essential sentences by finding the TF-IDF measure of each word in the sentence. Term Frequency (TF) is the frequency of each term in the sentence (its importance), and the Inverse Document Frequency (IDF) is the number of sentences in which that word appears, indicating how prevalent the word is. LSA is a statistical approach that creates a matrix with the keywords as rows and sentence numbers in which they occur as columns. Using lexical cohesion for summarization has been researched comparatively less. Lexical cohesion technique advances all these methodologies because of its concept of bringing together the text as a whole. It makes the resulting summary more readable as all the sentences are linked to each other.

2.3.1 Summarization using lexical cohesion

Following the theory of lexical cohesion by Halliday et al. (1976), many researchers have used lexical chains to summarize documents. Figure 2.2 lists some of the works that used lexical cohesion-based automatic summarization discussed below. The model for Lexical chains was first introduced by Morris and Hirst (1991). They assert that lexical chains are sequences of related words, and they share a distance relationship in each chain that they co-occur after a span (Morris and Hirst (1991)). Adapting this idea of lexical chains, Barzilay and Elhadad (2000) created a summarization system. The algorithm follows the creation of lexical chains of each noun in the text with their synonym, hyponym, hypernym, or siblings (using WordNet) and scoring them based on the length and homogeneity index (the measure of the distinct occurrence of words in the chain). The chains with a score greater than the average by two standard deviations are identified. The first member of each chain is selected, and the first sentence containing that word is extracted to form the summary. The limitations of their method include sentence granularity (long sentences have more tendency to be selected as every single unit is considered), and the summary length is not controllable.

Based on the work of Barzilay and Elhadad (2000), Silber and McCoy (2000) presents an efficient, linear-time algorithm that performs extractive summarization using lexical chains. The idea is to create the lexical chains and then create meta-chains of relationships in the text using them. A meta-chain contains a score and a list of words that form the chain. The score is calculated on the addition of each word to the chain. Each word in the text is inserted into the existing meta-chains if it matches the lexical chain of that word. Once the chains are formed, the meta-chain whose score will be most affected on the deletion of the word is identified for each word. The score of all the meta-chains in which the word is present is then adjusted. This process is repeated for all words in the text,

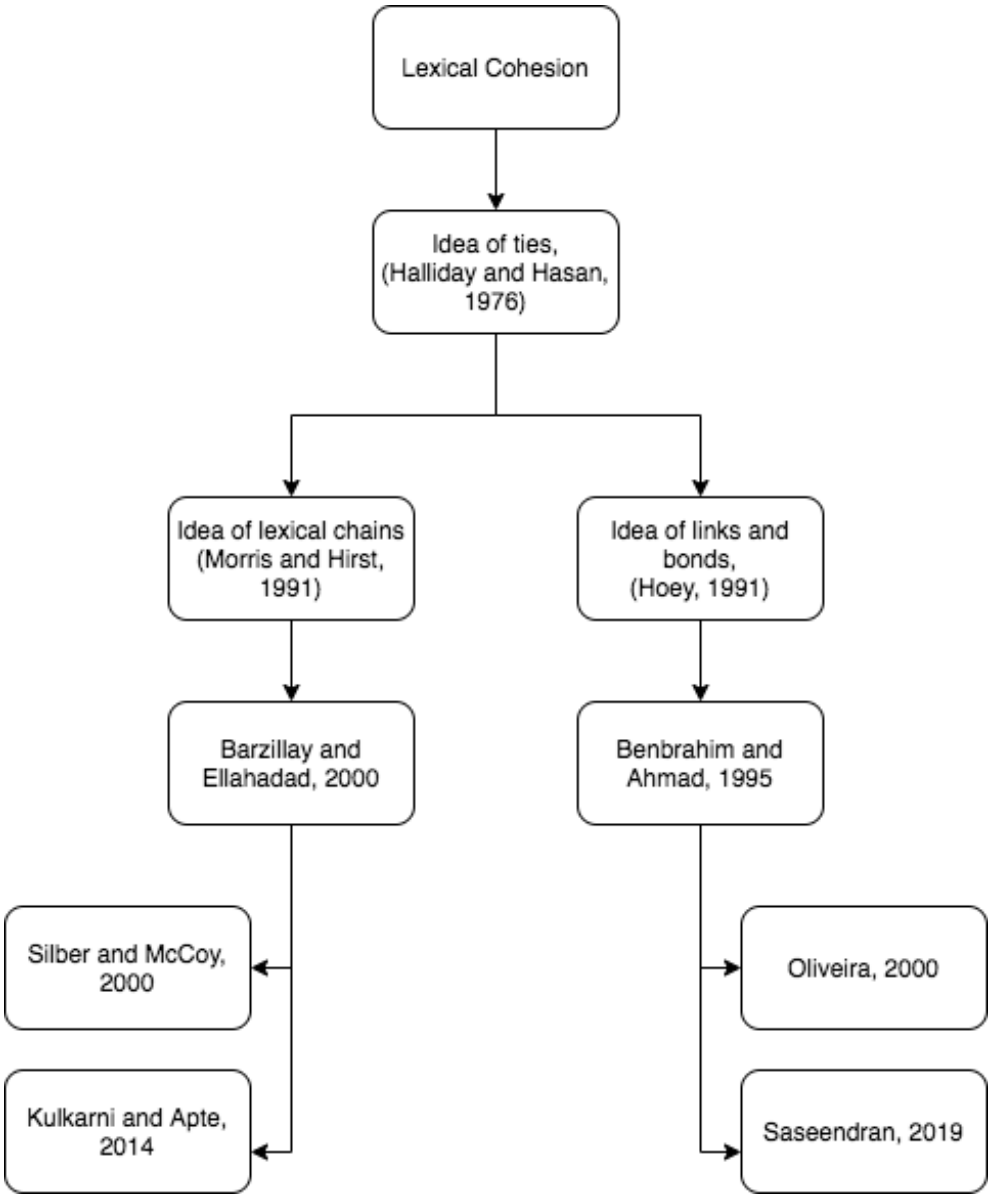

Figure 2.2: Few works on text summarization using lexical cohesion

and the chain with the highest score is selected. Though their method appears promising, the trials conducted were quite limited, and the evaluation of generated summaries was omitted.

Kulkarni and Apte (2014) added the concept of correlation of sentences on top of Barzilay and Elhadad (2000)'s algorithm to produce effective summaries. The process of creating lexical chains and identifying the chains with the highest score is included as part of their algorithm. Furthermore, from the selected sentences, those that begin with terms like although, however, this, those, and so on are seen to be related to the preceding statements. If the previous sentence's rank is more than or equal to 70% of the rank of the chosen sentence, the summary includes them. The summary generated was evaluated using precision and recall measures by comparing it to a manually extracted summary. They performed evaluation only on three brief texts, which does not offer the reader a clear picture of the effectiveness of their method.

Using lexical chains to understand the cohesion in the text is difficult and time-consuming. Instead, few researchers followed Hoey (1991)'s idea of creating a link matrix and bond matrix. TELE-PATTAN was one such summarization tool modeled by Benbrahim and Ahmad (1995). It uses a set of morphological rules and Macquarie's encyclopedic thesaurus to extract the lexical relations in the underlying text. From these relationships, a text map is constructed that depicts the links found in the text. According to Hoey (1991), there should be a minimum of three links between two sentences to accept them as bonded. This threshold level is termed the bond threshold. The text map with the links is converted to a bond network, and sentences are classified as topic opening, topic closing, and central sentences based on the density of the links (Benbrahim and Ahmad (1995)). TELE-PATTAN was designed as an interactive system that allowed users to set the bond threshold. It was practically better as some texts may have a high proportion of sentences with three repetitions (Hoey (1991)). The user could also choose the type of summary that was required (select only sentences from topic opening, topic closing, or central). The extracts were evaluated manually by four scientists based on the readability, content, and quality and achieved promising results.

Following on from TELE-PATTAN, de Oliveira et al. (2002) developed SummariserPort, a Java implementation of Hoey's two repetitions - simple repetitions and complex repetitions that could summarize financial news. For identifying complex repetitions, the algorithm used a preprocessed list of derivational suffixes. It included 75 morphological rules, which resulted in 2500 potential word relationships. The algorithm ignores a set of stopwords that contain the closed class words. It then creates a link matrix, indicating the number of links between the sentences based on the repetitions, and making use of this, the system models a bond matrix that confirms a bond for all the sentences with

more than three links. The sentences are then split and ranked into topic-opening, topic-closing, and central sentences based on the number of bonds to preceding and succeeding sentences. From each class, 10% of the sentences are extracted to create the summary, which accounts for 30% of the content. The authors evaluated the performance of their system by conducting a trial questionnaire to a group of PhD students and financial traders whether the summary contains all necessary information and excludes unnecessary information. They observed that the manual evaluation of the system's efficacy yielded different results among the evaluators, demonstrating that the judgment is solely dependent on the evaluator's expectations. It clearly shows the need for an automatic evaluation of summaries.

Saseedran (2019) extended the work of BenBrahim's TELE-PATTAN and created a system called *Curukka* that could summarize text from any discipline. The main contribution of his work was on the keyword identification for the creation of the link matrix. He proposed three methods like Weirdness Index, Part-Of-Speech (POS) tagging, and Collocation Analysis. The weirdness index score of a term in a text provides its significance compared to a reference corpus. It is the ratio of the relative frequency of the word in the reference corpus (Open American National Corpus and British National Corpus) and the text to be summarized. Stanford NLP POS-tagger is used to implement POS tagging and annotates each word in the text as a noun, conjunction, determiner, etc. The terms annotated as a noun is considered as a keyword. Collocation analysis recognizes compound words that appear together in a text. Once the keywords are identified, the rest of the summarizing process involves generating a link matrix and a bond matrix and then categorizing the sentences as topic opening, topic closing, or middle sentences. Saseedran (2019) also proposed the automatic evaluation of summaries based on the word distribution and the readability level of the extracted text. As for readability, he used the Flesch-Kinkaid Ease Reading formula that computes the readability level using the number of words, sentences, and syllables. In terms of word distribution, the relative frequency of all words in the summary is compared to the frequency of those words in the input text. In addition, the cumulative relative frequency of the top ten open class terms is compared.

2.3.2 Automatic summary evaluation

Traditionally, the evaluation of summarization systems was implemented manually where a group of intellectuals from the same discipline was assigned a questionnaire to judge the summary regarding coherence, conciseness, grammaticality, readability, and content (Mani (2001)). Manual work is never feasible. Hence, the need for automated evaluation

of summaries is high. The main difficulty in the evaluation process is that there is no clear definition as to what constitutes a good summary (de Oliveira (2005)). Another concern is that the evaluation would be limited to the evaluator's expectations.

An ideal summary should cover the relevant information from the input text and must omit the unnecessary information. According to Lloret et al. (2018), a summary is evaluated based on i) its language quality or readability, ii) its informativeness or content coverage, and iii) its non-redundancy. Previous works in automatic evaluation used a human-generated summary as the reference summary and calculated the metrics like precision, recall, and F-measure.

$$Recall(R) = \frac{human\text{-}generated\ summary\ sentences\ \cap\ automated\ summary\ sentences}{human\text{-}generated\ summary\ sentences} \tag{2.1}$$

$$Precision(P) = \frac{human\text{-}generated\ summary\ sentences\ \cap\ automated\ summary\ sentences}{automated\ summary\ sentences} \tag{2.2}$$

$$F\text{-}measure = 2 * \frac{R * P}{R + P} \tag{2.3}$$

The issue with this technique is that the scores are depended on the sentences chosen by the human. A sentence that expresses the same meaning but is not included in the human-generated summary reduces the total score of the system. To surmount this issue of subjective dependence on one human-generated summary, Radev and Tam (2003) proposed a metric called *Relative Utility*. This metric allows an evaluator panel to award a score of 0 to 10 to each sentence in the input text that they believe is relevant to the summary. The major demerit for this method is that scoring each sentence in the input text is tiresome and time-consuming. Papineni et al. (2002) introduced a metric called *BLEU*, an n-gram co-occurrence statistics measure that automatically evaluated machine translations based on a set of reference translations. The idea of BLEU was carried forward by Lin (2004), to introduce a package called Recall Oriented Understudy for Gisting Evaluation (ROUGE) that measured similarity between summaries. There are four measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S.

ROUGE-N is recall-related and compares the n-grams in the candidate and reference summaries. There can be multiple reference summaries. ROGUE-N is calculated as below.

$$ROGUE\text{-}N = \frac{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count(gram_n)} \tag{2.4}$$

where n is the length of the n-gram, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and reference summaries.

ROUGE-L is based on the longest common subsequence (LCS). It has various applications like that in molecular biology (DNA sequences) or file comparison. Here, the summary sentences are considered as a sequence of words and the longer LCS of two sentences makes them similar. The LCS-based F-measure proposed by Lin (2004) with X of length m as reference summary and Y of length n as candidate summary is,

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{2.5}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{2.6}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{2.7}$$

where LCS(X,Y) is the length of a longest common subsequence of X and Y, and β is the relative weighting factor between recall and precision.

ROUGE-W improves on the LCS method in case of consecutive matches in the sequence. A dynamic programming algorithm was implemented to solve this. The idea was to memorize the length of consecutive matches and form a dynamic table. The measure is calculated as below.

$$R_{wlcs} = f^{-1}(\frac{WLCS(X,Y)}{f(m)}) \tag{2.8}$$

$$P_{wlcs} = f^{-1}(\frac{WLCS(X,Y)}{f(n)}) \tag{2.9}$$

$$F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs}+\beta^2 P_{wlcs}}) \tag{2.10}$$

where $f^{-1}(k) = k^{\frac{1}{2}}$ and β is a weighting factor which assigns different relative importance to precision and recall.

ROUGE-S is a metric that measures the common skip-bigrams between the sentences. A skip-bigram is a combination of any pair of words in a sentence allowing gaps between them. For example, the sentence 'police killed the gunman' has the following skip-grams, {police killed, police the, police gunman, killed the, killed gunman, the gunman} (Lin (2004)). The calculation is as below.

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \tag{2.11}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \tag{2.12}$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2} + \beta^2 P_{skip2}} \tag{2.13}$$

where SKIP2(X,Y) is the number of skip-bigram matches between X and Y, β controls the relative importance of P_{skip2} and R_{skip2}, and C is the combination function.

ROUGE metrics have been used by many researchers to evaluate their summarization systems. It was also used during the Document Understanding Conference (DUC) 2004. Though it evaluates the effectiveness of summaries, there is a requirement for a human-generated summary as a reference. This dependency does not make it a fully automated solution. Therefore, there is a need for creating an automatic text summarization evaluation tool that can mimic the human judgment of the summary based on the input text alone.

2.4 Conclusion

In this chapter, we first dived into the background of the project, the issues related to fake news spread, and how the proposed method would aid in coping with it. The theory of lexical cohesion was then briefed and works on summarization systems using the concept of lexical cohesion, was discussed. Hoey's link matrix and bond matrix approach appears promising and is thus adapted in this project study. As for evaluation methodologies, existing methods were discovered to involve manual labor and are not automated completely.

Chapter 3

Methodology

3.1 Overview of the Approach

A large volume of text gets generated each day, and researchers are upgrading the methodologies to keep up with it. The wide variety of use cases of understanding the text have increased the importance of Natural Language Processing. Few use cases include grammar correction tools, text summarization, chatbots, sentiment analysis, recommendation systems, speech recognition, email classification. Analyzing texts have always faced the difficulty of unstructured and inappropriate data. Therefore, the data preprocessing step in a text analytics project is always tedious and requires some manual work. Another issue with text processing is that the process solely depends on the underlying language. A program that works on an English language text can not process in the same way on Japanese or Chinese text.

A wide variety of open source tools and corpora are available that helps in performing text analytics. Antconc is a corpus analysis toolkit for analyzing the text that is uploaded (Anthony (2011)). It has various options like concordance - search a keyword in the text, clusters - to search for a group of words, collocates to identify the nearby words, word list, and keyword list - most frequent words in the text based on a reference corpus. Drivel Defence is a software package tool developed by Plain English Campaign to check the use of plain English. There is a list of words considered to be plain, and advanced ones are mapped with plain. The Drivel Defence tool analyzes language in any document or even on a webpage and delivers statistics. It will specify the length of the sentences, give the average report length, suggest some substitution for words in the advanced word list. We also have an option to save the analysis results as a file. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a tool that helps in understanding how much positive or negative emotion the text has. The main advantage of this tool is that

it does not require training data and can perform on any domain. The Oxford dictionary created British National Corpus (BNC) in the 1980s and consists of 100 million words from multiple fields. ANC is a corpus containing American English text from 1990. It includes only those texts by authors who were born or educated and currently living in the US. It is annotated with part of speech, shallow parse annotations, and named entities. This project work makes use of some of the existing tools and is discussed in the upcoming sections.

Figure 3.1 below presents the steps followed by the text summarization method. The system is implemented in Python using various libraries like nltk, pattern, etc. The first step is to preprocess the text to be summarized. The next step is to extract keywords from the processed text. The text summarization algorithm employed is based on Hoey's link matrix and bond matrix method. The program detects keyword repetition in each sentence pair and generates the link matrix accordingly. Sentences having more than three links in the link matrix are regarded to have a bond. As a result, the bond matrix is created, and sentences are categorized as Opening, Closing, or Central sentences. The algorithm then extracts strongly bonded sentences from each category. As for evaluating the summarization system, statistical measures like Gunning Fog readability index, Sentiment score comparison, Cosine similarity, Jaccard similarity, and Pearson correlation of word distribution for syntactic comparison, are used. The coming section elaborates on the text summarization method, and the following section presents the evaluation techniques.

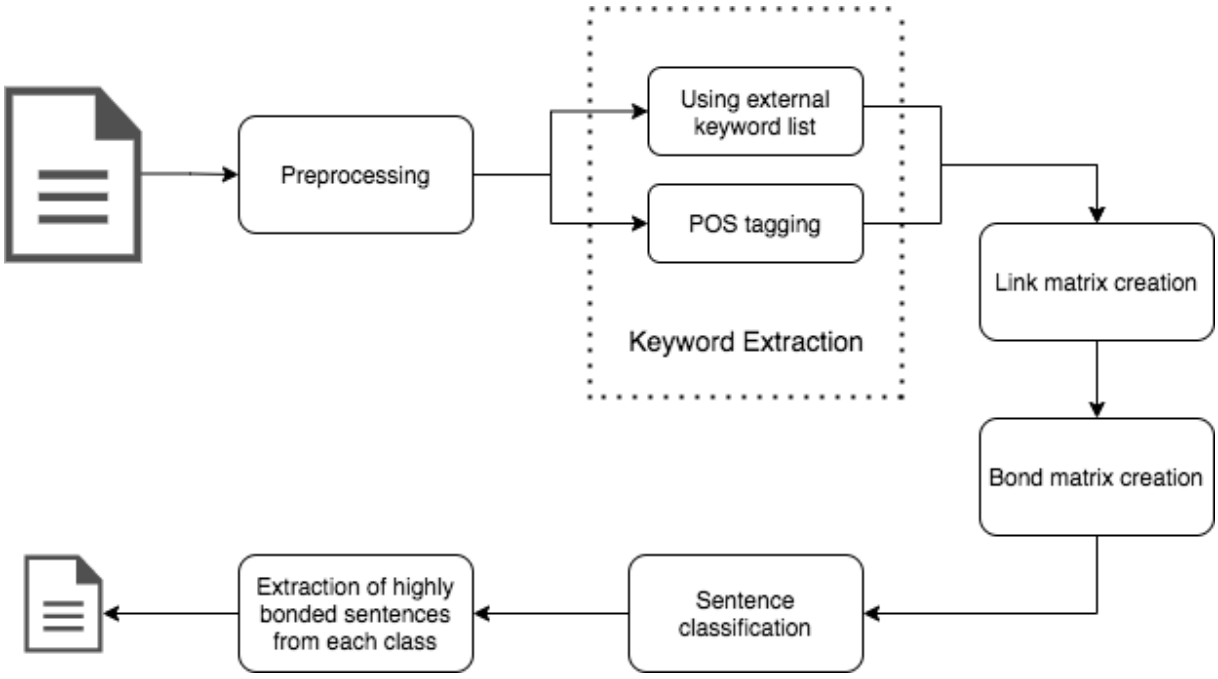

Figure 3.1: The flow diagram of the text summarizer

3.2 Text summarization using Lexical Cohesion

3.2.1 Preprocessing

The system developed can summarize a single document at once. The input text undergoes a preprocessing stage. During this stage, the input is first tokenized into sentences and then into words. The nltk library in Python helps in this process using the functions sent_tokenize() and word_tokenize(). Once the tokens are separated, the next task is to clean them. All the punctuations excluding hyphen (-) in each word are removed using *re library*, and letters are converted into lowercase. This process gives us a 2D list of cleaned tokens. The flow diagram (Figure 3.2) below illustrates the same.

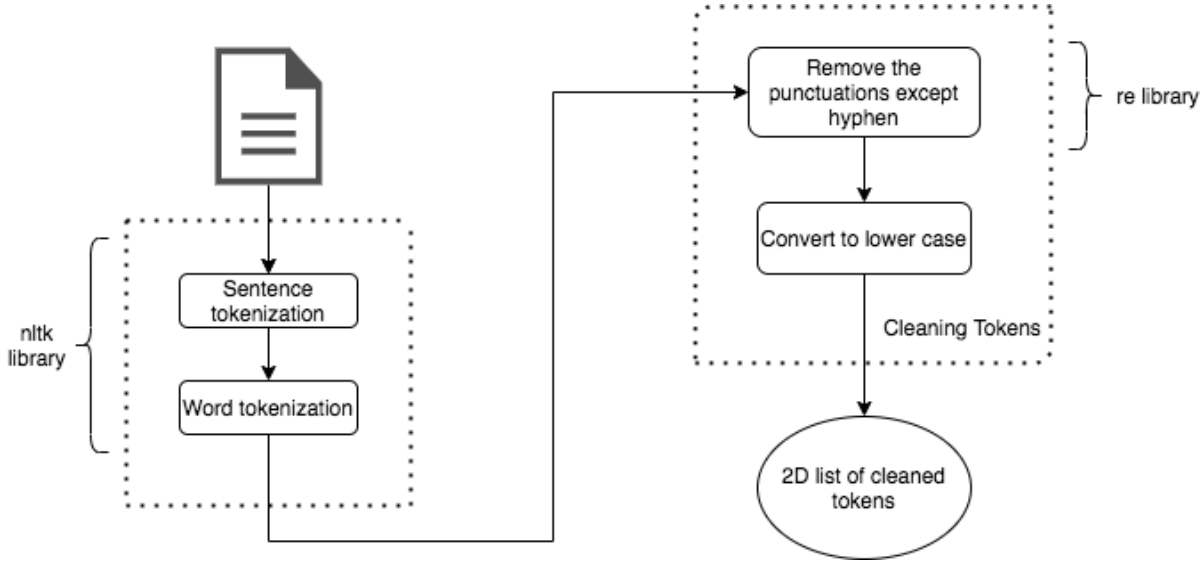

Figure 3.2: The flow diagram of the preprocessing stage

3.2.2 Keyword Extraction

Once all the tokens are cleaned, we now check whether it is a keyword. Keywords are identified using two methods: 1) Using a separate keyword list, 2) Using Parts-of-Speech tagging. The summarization system has the option to choose the method of keyword extraction.

Keyword list

Each token is compared to a list of keywords present in the coronavirus dataset relative to a reference corpus. The keyword list was created using a corpus management tool called Sketch Engine (Kilgarriff et al. (2014)). Sketch Engine is a corpus query tool

managed by Lexical Computing that is used by linguists worldwide. It provides multiple corpora in 90+ different languages and has various features like word sketch to analyze the collocations, thesaurus to find synonyms and antonyms, concordance analysis, keyword list, etc. The keyword list feature assists in finding the unique or typical terms found in a corpus relative to a reference corpus. The coronavirus corpus consists of texts released as part of the COVID-19 Open Research Dataset (CORD-19). The reference corpus used is English Web 2013 (enTenTen13) that consists of 19 billion words. The relative frequency of terms in each corpus is compared to identify the keywords (a list of top 1000 words). Table 3.1 showcases a few of the keywords present in the list.

Table 3.1: Top 20 keywords in coronavirus dataset according to SketchEngine

Sl. No	Keyword
1	sars-cov
2	rna
3	mers-cov
4	coronavirus
5	sars
6	usepackage
7	pcr
8	influenza
9	titer
10	μl
11	pedv
12	ifn
13	viral
14	assay
15	antiviral
16	mrna
17	rt-pcr
18	rsv
19	epitope
20	antibody

Analyzing this list indicated that it lacks named entities such as the names of firms that made vaccinations, the names of vaccines, and scientists investigating the virus. These have also been added to the list as they are significant in the context of the coronavirus dataset. Table 3.2 shows few terms added to the original keyword list. After the preprocessing stage, the cleaned tokens are searched on the keyword list and, if found, are included as filtered tokens.

Table 3.2: Few named entities added to the keyword list

Sl. No	Keyword
1	pfizer-biontech
2	moderna
3	novavax
4	astrazeneca
5	johnson
6	sanofi
7	glaxosmithkline
8	curevac
9	merck
10	roche
11	comirnaty
12	tocilizumab
13	Koopmans
14	Nguyen
15	Daszak
16	Kariko
17	Gilbert
18	Pollard
19	Zaks
20	Moore

Part-Of-Speech Tagging

Part-of-speech tagging (POS tagging) is the method of marking up a word in a text (corpus) as corresponding to a specific part of speech, depending on both its meaning and its context (its relationship with surrounding words in the sentence). Anyone learning a new language is commonly asked to recognize word classifications such as nouns, verbs, adjectives, etc. It is the same procedure as POS tagging. Python's nltk package enables POS tagging. Table 3.3 displays the tags issued by a POS tagger. The cleaned tokens are passed to the pos_tag function of nltk to get a dictionary of tokens with their corresponding tag. All the tokens with a tag starting with 'NN' indicating it as a noun, is filtered out as keyword.

Table 3.3: POS tags in nltk with examples

No.	Tag	Description	Example
1.	CC	Coordinating conjunction	and, but
2.	CD	Cardinal number	digit 1
3.	DT	Determiner	a, an, the
4.	EX	Existential there	there exists
5.	FW	Foreign word	words in language other than main text
6.	IN	Preposition or subordinating conjunction	in, on
7.	JJ	Adjective	large
8.	JJR	Adjective, comparative	larger
9.	JJS	Adjective, superlative	largest
10.	LS	List item marker	1)
11.	MD	Modal	could, will
12.	NN	Noun, singular or mass	tree
13.	NNS	Noun, plural	trees
14.	NNP	Proper noun, singular	Mary
15.	NNPS	Proper noun, plural	Indians
16.	PDT	Predeterminer	all, both
17.	POS	Possessive ending	parent's
18.	PRP	Personal pronoun	I, he, she
19.	PRP$	Possessive pronoun	my, his, her
20.	RB	Adverb	good
21.	RBR	Adverb, comparative	better
22.	RBS	Adverb, superlative	best
23.	RP	Particle	give up
25.	TO	*to*	to
26.	UH	Interjection	Mmm
27.	VB	Verb, base form	take
28.	VBD	Verb, past tense	took
29.	VBG	Verb, gerund or present participle	taking
30.	VBN	Verb, past participle	taken
31.	VBP	Verb, non-3rd person singular present	take
32.	VBZ	Verb, 3rd person singular present	takes
33.	WDT	Wh-determiner	which
34.	WP	Wh-pronoun	what
35.	WP$	Possessive wh-pronoun	whose
36.	WRB	Wh-adverb	when

3.2.3 Repetition analysis or Link matrix creation

Once the keywords are identified, the next step is to find the repetitions. The system implements all four classes of repetitions specified by Hoey (1991). Figure 3.3 depicts the flow diagram of repetition analysis.

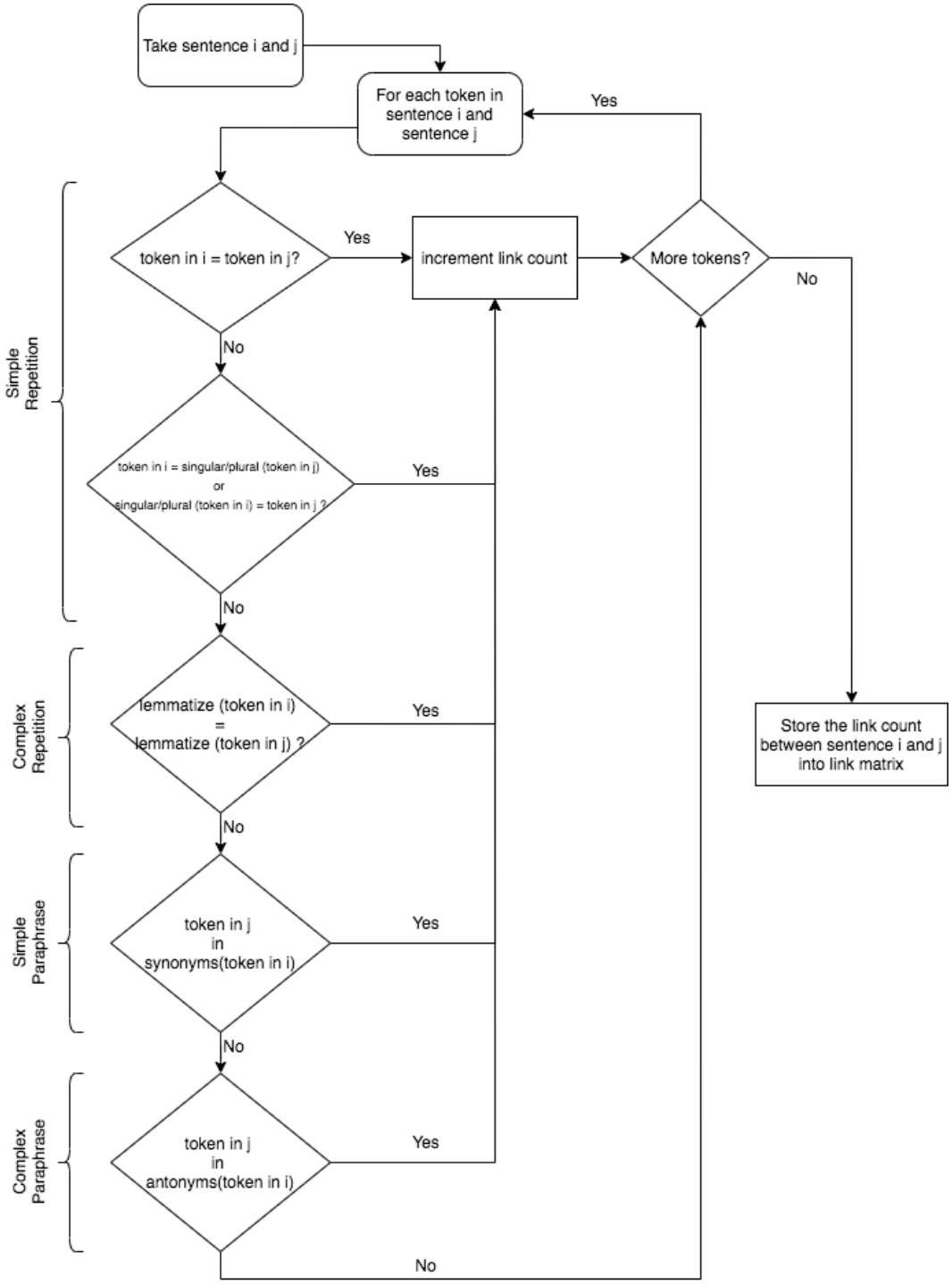

Figure 3.3: The flow diagram of link matrix creation

Simple Repetition

Simple Repetition occurs whenever the exact word or the singular/plural form of it is repeated. The pattern library of Python has two functions, singularize and pluralize, that give the singular and plural form of the word supplied. The 2D array of filtered tokens in each sentence is iterated first to check for simple repetition between each word in a sentence pair. Each repetition between a sentence pair would increment the link count and is reflected in the link matrix.

Complex Repetition

Complex Repetition occurs whenever the lexeme is repeated in the text. Words like 'drug', 'drugging', and 'drugged' are examples of this form of repetition because they have the same lexeme - 'drug'. To check the complex repetition between each word in a sentence pair, we use lemmatization. The technique of collecting together the various variant forms of a word to study them as a single item is known as lemmatization. The nltk provides the WordNetLemmatizer module that would lemmatize any word supplied. Once the words are compared for simple repetition, they are converted or lemmatized to their base form and compared for complex repetition. Like in simple repetition, a match increases the link count and is recorded in the link matrix.

Simple Paraphrase

A Simple paraphrase in a text indicates the existence of synonyms among the sentences. The WordNet module from nltk is used to search synonyms of a word in other sentences. Synset is the interface provided by WordNet for obtaining synonymous words. We generate a list of synonyms for each word in the 2D array of filtered tokens (sentence 1) that we traverse over. The corresponding link is incremented if the word in the other sentence (sentence 2) is in the list of synonyms. Simple paraphrase checking is used only if simple repetition or complicated repetition does not yield a match.

Complex Paraphrase

Complex Paraphrase occurs in two scenarios. The first case is the existence of antonyms among the sentences. The second case is when there is a combination of simple paraphrasing and complex repetition (See Figure 3.4). In our method, only the first case of complex paraphrasing is implemented. In this scenario, the WordNet module comes in handy, just as it does for synonyms. It provides a set of antonyms of a word passed to it.

Each word in sentence two is checked against the list of antonyms, and if found, the link count is incremented.

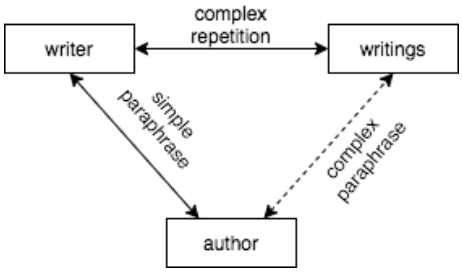

Figure 3.4: Complex paraphrase scenario 2 (Hoey (1991))

The algorithm followed for creating the link matrix is as below (Algorithm 1).

Algorithm 1: Link matrix creation

```
Data: sentences of length n
Result: link_matrix of size n x n
for i in range(n) do
    for j in range(i+1 to n) do
        ct ← 0;
        for w1 in cleaned_sentences[i] do
            for w2 in cleaned_sentences[j] do
                if w1 == w2 then
                    ct++ ;
                else if w1 == singularize(w2) or singularize(w1) == w2 then
                    ct++ ;
                else if w1 == pluralize(w2) or pluralize(w1) == w2 then
                    ct++ ;
                else if lemmatize(w1) == lemmatize(w2) then
                    ct++ ;
                else if w2 in getsynonym(w1) then
                    ct++ ;
                else if w2 in getantonym(w1) then
                    ct++ ;
                else
                    do nothing;
                end
            end
            link_matrix[i][j] ← ct;
            link_matrix[j][i] ← ct;
        end
    end
end
```

An example of links between few sentences in a text is presented in Figure 3.5. The bolded words in each sentence correspond to the keywords identified using the external keyword list. Sentences 25 and 26, as well as Sentences 25 and 28, share a common link as a result of simple repetition due to the occurrence of the word 'vaccine' in both sentences. Sentences 26 and 28 have four links because of the simple repetition of the words: 'mRNA,' 'CureVac,' and 'vaccine.'

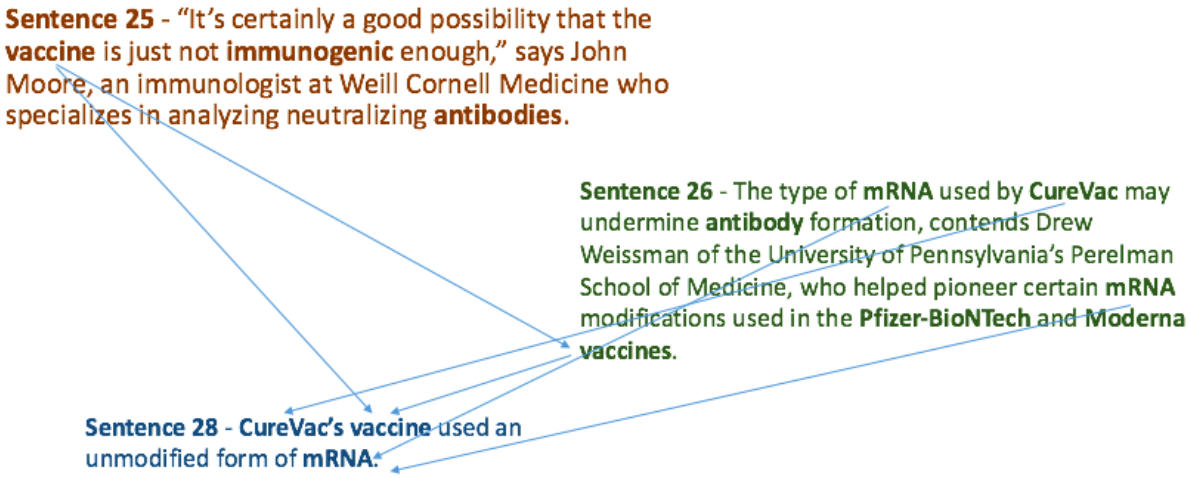

Figure 3.5: An example of keyword extraction and sentence linking

Table 3.4 presents an extract of the link matrix for the same input text with 47 sentences (Sentence from 25 to 45 only shown here due to space constraints). The sentences 25 and 28 previously discussed, have a link value of 4 in the matrix.

3.2.4 Bond matrix creation

Once the links between each sentence pair are identified, the next step is to create a bond matrix that would depict whether two sentences are bonded. The threshold limit, the number of links required between two sentences for them to be bonded is called the bond strength. According to Heoy's theory, there should be a minimum of 3 links between sentences for them to be bonded. He also claims that the strength of this relationship depends on the text under consideration. For example, if more than 75% of the sentences have more than three links between them, fixing the bond strength as three would be ineffective as we cannot capture the most important sentences. For such a case, the bond strength should be set high. But for simplicity, we consider the bond strength to be 3. The link matrix from the previous step is iterated to produce the bond matrix. If the link equals three or more, the corresponding cell will have a value of 1 signifying a bond,

Table 3.4: An extract of link matrix of a sample text with 47 sentences

	S25	**S26**	**S27**	**S28**	**S29**	**S30**	**S31**	**S32**	**S33**	**S34**	**S35**	**S36**	**S37**	**S38**	**S39**	**S40**	**S41**	**S42**	**S43**	**S44**	**S45**
S1	1	2	0	2	0	1	0	0	0	1	2	0	1	0	0	1	1	1	2	1	**3**
S2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	1	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1
S4	0	3	0	2	1	1	0	1	1	1	1	0	1	0	0	1	0	0	2	1	1
S5	1	**5**	0	2	1	0	0	1	1	2	1	0	0	0	0	0	**3**	1	1	**3**	2
S6	1	2	0	2	0	1	0	0	0	1	2	0	1	0	0	1	1	1	2	1	3
S7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S10	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1
S11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S12	1	**3**	0	2	1	0	0	0	1	2	1	0	0	0	0	0	1	1	1	2	1
S13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S15	1	3	0	2	1	0	0	0	1	2	1	0	0	0	0	0	1	1	1	2	1
S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S17	2	**3**	0	**3**	0	1	0	0	0	2	**3**	0	1	0	0	1	2	2	**3**	2	**5**
S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S19	1	2	0	2	0	1	0	0	0	1	2	0	1	0	0	1	1	1	2	1	2
S20	1	**6**	0	**3**	1	1	0	1	1	2	2	0	1	0	0	1	3	1	2	**3**	2
S21	1	2	0	2	0	1	0	0	0	1	2	0	1	0	0	1	1	1	2	1	3
S22	1	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	2
S23	1	2	0	2	0	1	0	0	0	1	2	0	1	0	0	1	1	1	2	1	2
S24	2	2	0	1	0	0	**3**	0	1	2	1	0	0	0	0	1	1	1	**3**	2	1
S25	0	2	0	1	0	0	1	0	1	2	1	0	0	0	0	1	1	1	2	2	1
S26		0	0	**4**	2	1	1	1	3	**4**	2	0	1	0	0	2	**3**	1	**3**	**5**	2
S27			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S28				0	1	1	0	0	1	2	2	0	1	0	0	1	1	1	2	2	2
S29					0	0	1	0	2	1	0	0	1	0	0	0	0	0	0	1	0
S30						0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1
S31							0	0	1	1	0	0	0	0	0	1	0	0	**3**	1	0
S32								0	0	0	0	0	0	0	0	0	1	0	0	1	0
S33									0	2	0	0	1	0	0	1	0	0	1	2	0
S34										0	1	0	0	0	0	1	1	1	2	**3**	1
S35											0	0	1	0	0	1	1	1	2	1	2
S36												0	0	0	0	0	0	0	0	0	0
S37													0	0	0	1	0	0	1	0	1
S38														0	0	0	0	0	0	0	0
S39															0	0	0	0	0	0	0
S40																0	0	0	2	1	1
S41																	0	1	1	2	1
S42																		0	1	1	1
S43																			0	2	2
S44																				0	1
S45																					0

Table 3.5: An extract of bond matrix of a sample text with 47 sentences

	S25	**S26**	**S27**	**S28**	**S29**	**S30**	**S31**	**S32**	**S33**	**S34**	**S35**	**S36**	**S37**	**S38**	**S39**	**S40**	**S41**	**S42**	**S43**	**S44**	**S45**
S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
S6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S17	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S20	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
S21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S24	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S26		0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0
S27			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S28				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S29					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S30						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S31							0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S32								0	0	0	0	0	0	0	0	0	0	0	0	0	0
S33									0	0	0	0	0	0	0	0	0	0	0	0	0
S34										0	0	0	0	0	0	0	0	0	0	1	0
S35											0	0	0	0	0	0	0	0	0	0	0
S36												0	0	0	0	0	0	0	0	0	0
S37													0	0	0	0	0	0	0	0	0
S38														0	0	0	0	0	0	0	0
S39															0	0	0	0	0	0	0
S40																0	0	0	0	0	0
S41																	0	0	0	0	0
S42																		0	0	0	0
S43																			0	0	0
S44																				0	0
S45																					0
	0	**6**	0	3	0	0	1	0	1	1	1	0	0	0	0	0	3	0	**4**	**4**	**4**

otherwise 0. The bond matrix generated corresponding to the link matrix in Table 3.4 is presented in Table 3.5.

3.2.5 Classifying sentences as Topic Opening, Topic Closing, Middle and Marginal

According to Hoey, the marginal sentences are those with very few or no bonds. These sentences are less coherent than other sentences and convey little information. They are present in the text to make it more readable. Central sentences are those with a high level of bonding compared to other sentences. If we manually analyze the number of sentences with each bond value, we can deduce the threshold value that can be set for them to be

central. This threshold value differs in each text and is not easy to find it automatically. For simplicity, all the sentences with a bond value greater than three are identified as important sentences. The important sentences can be classified as topic opening and topic closing based on the number of bonds it has to the preceding and succeeding sentences. If there are more bonds to the succeeding sentence than the preceding, it is a topic opening sentence. In contrast, if the number of bonds to the preceding sentences is greater than the number of succeeding sentences, it is a topic closing sentence. A middle sentence has an equal number of links to both preceding and succeeding sentences. Following this theory, the sentences are classified as marginal, topic opening, topic closing, and middle sentences.

A column sum of all columns from the bond matrix would yield the number of bonds each sentence has with the other sentences. The average bond value of the text is calculated by adding all bond values and dividing by the total number of sentences. The threshold limit for a sentence to be considered important is set to 3. Any sentence with a bond value less than 3 is marginal. The sentences with a bond value greater than three are classified as important. To readily categorize the important sentences as topic opening, middle, and topic closing, the bond matrix is used to generate a bond tuple list with two entries and length as sentence count of the input text. The number of succeeding and preceding sentences for each sentence is counted and added to the tuple list. The categorization is implemented as below.

$$\text{Topic Opening} \leftarrow \left(\begin{array}{l} \text{No. of bonds with succeeding sentences -} \\ \text{No. of bonds with preceding sentences} \end{array} \right) < \textit{avg bond value}$$

$$\text{Topic Closing} \leftarrow \left(\begin{array}{l} \text{No. of bonds with preceding sentences -} \\ \text{No of bonds with succeeding sentences} \end{array} \right) < \textit{avg bond value}$$

Middle $\leftarrow$ Remaining important sentences

Table 3.6 presents the bond tuple and sentence classification of the sample text used earlier for link matrix and bond matrix creation. T.O indicates Topic opening, T.C indicates Topic closing, M indicates middle sentence.

Table 3.6: Sentence classification of the sample text

Sentence	Bond Tuple	Sentence Class	Sentence	Bond Tuple	Sentence Class	Sentence	Bond Tuple	Sentence Class
1	(0, 2)		17	(3, 11)	**T.O**	33	(1, 0)	
2	(0, 0)		18	(0, 0)		34	(1, 1)	
3	(0, 0)		19	(1, 0)		35	(1, 0)	
4	(0, 1)		20	(2, 4)	**T.O**	36	(0, 0)	
5	(0, 5)	**T.O**	21	(2, 1)		37	(0, 0)	
6	(0, 3)		22	(1, 0)		38	(0, 0)	
7	(0, 0)		23	(1, 0)		39	(0, 0)	
8	(0, 0)		24	(1, 2)		40	(0, 0)	
9	(0, 0)		25	(0, 0)		41	(3, 0)	
10	(0, 0)		26	(6, 6)	**M**	42	(0, 0)	
11	(0, 0)		27	(0, 0)		43	(4, 0)	**T.C**
12	(0, 1)		28	(3, 0)		44	(4, 0)	**T.C**
13	(0, 0)		29	(0, 0)		45	(4, 0)	**T.C**
14	(0, 0)		30	(0, 0)		46	(0, 0)	
15	(0, 1)		31	(1, 1)		47	(0, 0)	
16	(0, 0)		32	(0, 0)				

3.2.6 Sentence extraction for summary

Depending on the type of text summarized, the sentences can be extracted differently. Hoey proposed multiple abridgment procedures. The first method eliminates all marginal sentences. Another method includes all of the central sentences. The third one identifies the highly bonded sentence and includes all of the sentences that are bonded with it. Another method is to include topic opening and topic closing sentences. In this project, the research articles are summarized. So it is important to have a combination of topic opening, topic closing, and middle sentences. Including all sentences from these categories can produce long summaries and would not serve its purpose. Therefore, a limit for the number of sentences in the output summary is imposed. From each category, most bonded ones are selected. For input text with more than 100 sentences, a maximum of 20 sentences (6 topic opening, 6 topic closing, and 8 middle sentences) and input text, less than 100 sentences, a maximum of 10 sentences (3 topic opening, 3 topic closing, and 4 middle sentences) are extracted. The extracted sentences are sorted in the same order as in the input text to obtain the summary.

3.3 Automatic Summary Evaluation techniques

As discussed in the previous chapter, there is no clear-cut definition for a good summary. A summary can be tagged as 'effective' if it is readable, concise, include all necessary content, and conveys the same idea as the input text. A statistical evaluation of summaries based on readability, lexical similarity, syntactic similarity, and sentiment similarity is performed.

3.3.1 Gunning Fog Readability Index

The Gunning fog index is a readability measure for English text in linguistics. Robert Gunning, an American businessman who had previously worked in newspaper and textbook publishing, created the test in 1952 (Gunning (1952)). The index calculates the number of years of formal education required to grasp the material on the first reading. The fog index measure ranges from 6 to 17 (See Table 3.7 for the reading level of each index value).

Table 3.7: Gunning Fog index reading level

Fog Index	**Reading level by grade**
17	College graduate
16	College senior
15	College junior
14	College sophomore
13	College freshman
12	High school senior
11	High school junior
10	High school sophomore
9	High school freshman
8	Eighth grade
7	Seventh grade
6	Sixth grade

It calculates the word count in each sentence and the ratio of complex words to total words. Complex words consist of more than three syllables but exclude proper nouns, compound words, and common suffixes (like -es, -ed, -ing). The formula used in calculating Gunning Fog is as below.

$$Gunning\ fog\ index = 0.4[\frac{words}{sentences} + 100(\frac{complex\ words}{words})] \tag{3.1}$$

The Python Textstat library has a method for quickly computing the gunning fox index. By comparing the readability index of the summaries to the original text and the abstract, we can understand how much readable the generated summary is.

3.3.2 Sentiment similarity

The main aim behind this project work is to reduce the level of misrepresentation. A good summary should therefore capture the essence of the original text. Sentiment analysis or opinion mining helps to understand the positivity or negativity in the text. It is mostly used in customer review analysis for recommendation systems. If the sentiment of the

input text matches that of the summary, we can conclude that both convey either positive, negative, or neutral content.

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a text sentiment analysis tool that is sensitive to both emotion polarity (positive/negative) and intensity (strength) (Hutto and Gilbert (2014)). It is included in the NLTK package and can be used on unlabeled text data directly. VADER sentimental analysis is based on a lexicon that maps lexical characteristics to emotion intensities, which are known as sentiment scores. A text's sentiment score is calculated by adding the intensity of each word in the text. The result of any text analyzed is a dictionary of positive, negative, neutral, and compound sentiment scores. The compound score is the normalized measure of all the other scores. This measure thus helps compare the texts easily. This package is used to get the sentiment of the text and the summary.

3.3.3 Syntactic similarity

The main idea of syntactic similarity in text similarity is to compare texts based on the words they include. This comparison is carried out using three distinct methods: Cosine similarity, Jaccard similarity, and word distribution.

Cosine similarity

Cosine similarity represents the documents as vectors in a 2D space and finds the angle between them. If both the vectors point towards the same direction, it indicates they are similar. The documents are represented as vectors based on the unique words present in them. A term frequency vector indicating the count of each word in the document is created. The dot product between the two vectors gives the measure of cosine similarity. The value ranges from 0 to 1, with '1' indicating that the documents are identical. The measure is calculated as below:

$$Cosine\ similarity(a, b) = cos\ \theta = \frac{a.b}{|a||b|} \tag{3.2}$$

Jaccard Similarity

Jaccard index or Jaccard Similarity Coefficient was developed by Paul Jaccard, to compare the similarity between sample sets. It counts the number of unique items present in set A and set B (intersection), then divides that number by the union of unique terms in set A and set B.

$$Jaccard\ index(A, B) = \frac{A \cup B}{A \cap B} \tag{3.3}$$

Jaccard index can be used to find the syntactic similarity between two documents. First, we create two sets of unique words in each document. Then using the equation 3.3 we find the Jaccard index. The value ranges from 0 to 1 (0.5 means 50% similarity).

3.3.4 Pearson Correlation Coefficient of relative word frequency (Word distribution)

This measure is used to check the lexical similarity between the summary and input text, and the summary and abstract. It ensures that the extracted summary captures all of the necessary information. A correlation of relative frequency of keywords identified using POS tagging, in the text and the summary is determined. This measure provides us an understanding of how well the extracted summary captures relevant information. A high value indicates that the summary includes the required information. Pearson correlation coefficient gives a measure of the linear correlation between two sets of data. The calculation is as below.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{3.4}$$

where r = correlation coefficient, x_i = values of the x-variable in a sample,
$\bar{x}$ = mean of the values of the x-variable, y_i = values of the y-variable in a sample,
$\bar{y}$ = mean of the values of the y-variable

The Scipy library of Python has a pearsonr module that provides the function to get the Pearson correlation of two items. Algorithm 2 presents the steps in finding the relative frequency of keyword comparison.

Algorithm 2: Pearson correlation coefficient of relative frequency of words

Data: text1,text2
Result: Pearson correlation coefficient of relative frequency of words
$word\_count1 \leftarrow getWordCount(text1)$
$word\_count2 \leftarrow getWordCount(text2)$
for *word in word_count2* **do**
    if *word in word_count1* **then**
        $data1$.append($word\_count1[word]$)
    else
        $data1$.append(0)
    end
    $data2$.append($word\_count2[word]$)
end
$corr \leftarrow pearsonr$(data1,data2)

Chapter 4

Experiments and Results

4.1 Introduction

This chapter presents the experiments conducted to evaluate the method of text summarization detailed in the previous chapter. The summaries generated also undergo the automatic summary evaluation techniques. The upcoming section covers the details of the dataset used to perform the experiments. We implement summarization using both the external keyword list as well as the POS tagging method. The results of the evaluation measures for both techniques are presented. Various level of comparison is implemented that includes the extracted summary and the original text, and original text and the abstract (comparison to human-generated summary).

4.2 Dataset

Since the motivation for this project work was the requirement for accurate information during an unfamiliar atmosphere like the Covid-19 pandemic, the texts associated with coronavirus are utilized for experimentation. The project aims to eliminate misrepresentation in the initial level of the text cline. Therefore, articles related to coronavirus in popular science magazines like *The Scientist* and *Science* are scraped, and the research papers referred by these articles are also chosen. The idea behind such a selection is to showcase how different authors of science magazine articles articulate the information found in research publications. The final dataset consists of ten articles (3 articles from *Science* and 7 articles from *The Scientist*) and ten research papers referred by these articles. The articles are selected such that they all referred to at most one research paper to facilitate easy comparison. The chosen magazine articles and research papers are listed in the Appendix.

4.3 Summarization

The summaries are generated for all ten articles and ten research papers using the two keyword extraction methods: external keyword list and POS tagging. The bond strength is set as 3 (there should be a minimum of 3 links between sentences to tag them as bonded). If the input text has more than 100 sentences, then 20 sentences are extracted from it. This limit ensures that the summary produced is concise and contains only the most valuable information. For the input text with less than 100 sentences, ten sentences are extracted. As for summarization utilizing the external keyword list method, only three of ten science journal articles could be summarized, whereas the POS tagging method could summarize eight articles. Both methods were able to summarize all ten research papers. The summaries generated are evaluated using the techniques described in the previous chapter. The results obtained are showcased in the next section.

4.4 Evaluation Results

A good summary must be readable, be concise, include all pertinent information, and exclude non-relevant details. The statistical measures used to verify these criteria were discussed in the preceding chapter. Evaluation is performed by comparing the summary generated with the original text and the abstract (human-generated summary) where available. All research papers in the dataset have an abstract for comparison.

4.4.1 Sentence count and word count

Figure 4.1 and Figure 4.2 presents the sentence and word count of the ten research papers, the abstracts, and summaries from both keyword extraction methods. Analyzing the sentence count plot, we find that the summaries generated by both techniques are slightly longer than the abstract of the research paper. On average, the abstract is 7.5% of the input text. The summary produced by the external keyword list method is 10%, and the POS tagging method is 12%. There are few instances, such as in research paper 6, where the abstract is lengthier than the summaries. Because the sentence count limit was set at 10 for any input text containing fewer than 100 sentences and 20 for text containing more than 100 sentences, comparing the sentence count with the abstract makes little sense. But the understanding this difference and comparing the word counts is logical. The word count of abstracts is around 5.4 percent on average, whereas the summary generated using the external keyword list approach is 16.2 percent on average, and the POS tagging method is 18.6 percent on average. As compared to the sentence count, the

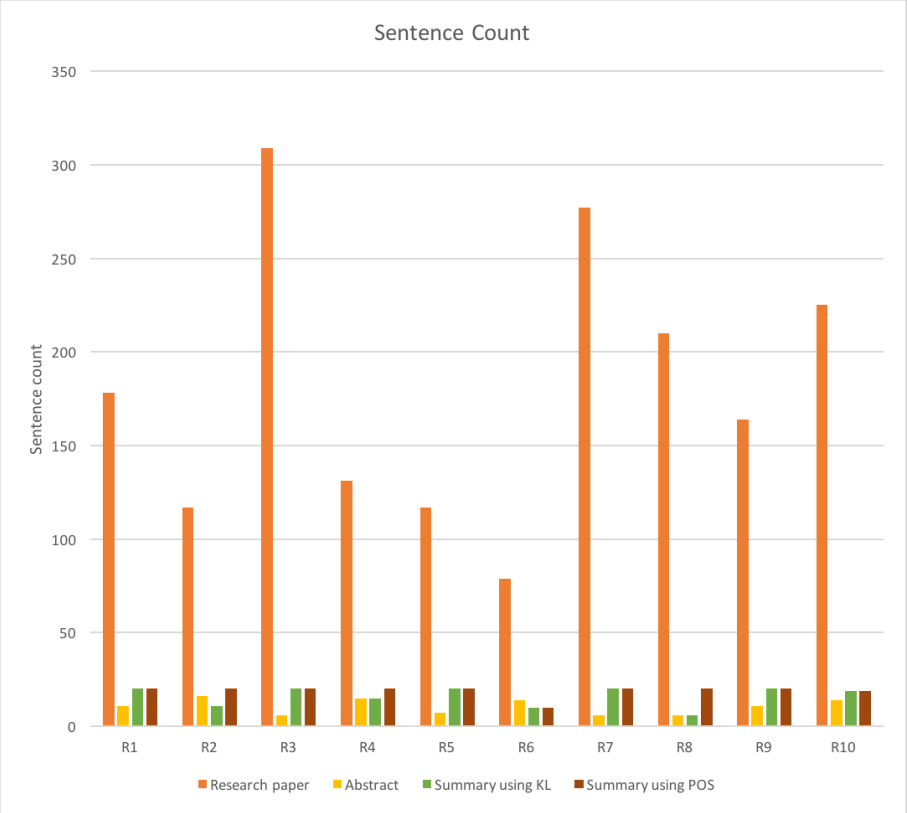

Figure 4.1: The sentence count of the research papers, their abstracts, and the summaries generated by the external keyword list and the POS tagging methods

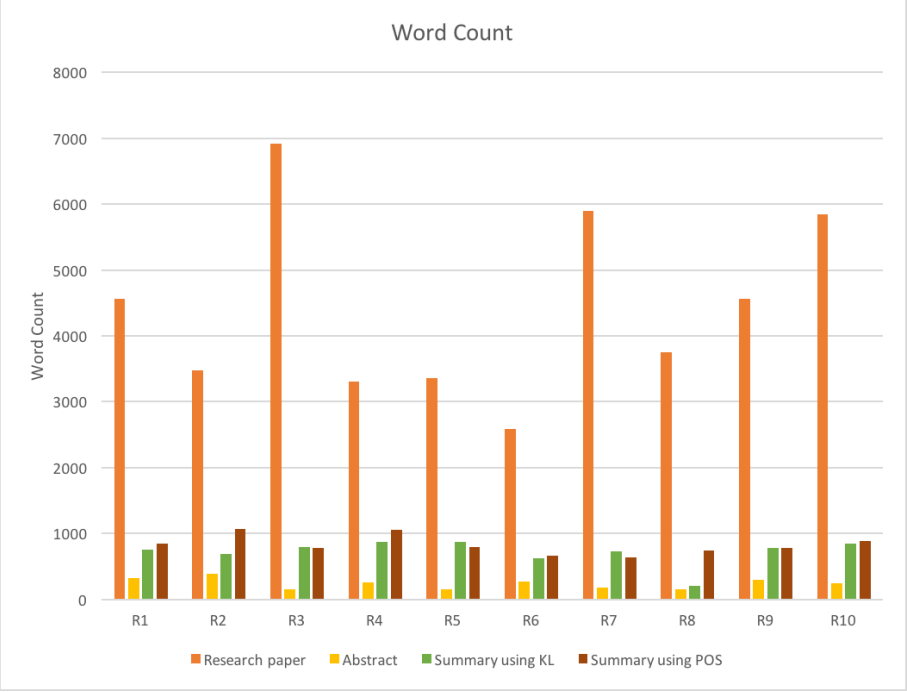

Figure 4.2: The word count of the research papers, their abstracts, and the summaries generated by the external keyword list and the POS tagging methods

word count difference is more. Table 4.1 shows the average sentence count, average word count, and average sentence length of the ten research articles, their abstracts, and the summaries produced. The average sentence length of the abstracts is less than the actual research papers. However, the average sentence length of the created summaries is nearly double that of the original research paper, indicating that the sentences extracted from the input text are longer than the remaining text. It would mean that the readability of the summaries is less compared to the abstract. To confirm this variation, we perform the gunning fog readability test (See Section 4.4.2).

Table 4.1: Average statistics of sentence count and word count in research papers, abstracts and the summaries

	Research paper	**Abstract**	**Summary using KL**	**Summary using POS**
Total sentence count	1807	106	161	189
Average sentence count	180.7	10.6	16.1	18.9
Total word count	44271	2409	7157	8247
Average word count	4427.1	240.9	715.7	824.7
Average sentence length	**24.5**	**22.73**	**44.45**	**43.63**

Table 4.2: Average statistics of sentence count and word count in articles, and the summaries

Article	**Sentence count**			**Word count**		
	Article	**Summary using KL**	**Summary using POS**	**Article**	**Summary using KL**	**Summary using POS**
A1	46	8	10	1080	240	334
A2	35	-	7	891	-	199
A3	68	1	10	1975	53	403
A4	71	-	8	1625	-	314
A5	34	10	9	905	314	303
A6	14	-	-	421	-	-
A7	7	-	-	696	-	-
A8	58	-	7	1338	-	217
A9	49	-	5	1070	-	133
A10	38	-	6	1117	-	328
Total	420	19	62	11118	607	2231
Average	**42.0**	**6.3**	**7.8**	**1111.8**	**202.3**	**278.9**
Standard Deviation	21.0	4.7	1.8	447.7	134.5	88.0

Table 4.2 presents the sentence count and word count of the science magazine articles summarized. Both techniques failed to summarize two articles: A6 and A7. The

external keyword list method could summarize only three articles. The results show that summaries created using the external keyword list technique have an average sentence count of 15.1% of the input text, while the POS tagging method has an average sentence count of 18.4% of the input text. The word count of the summaries of both techniques is 18.2% and 25.1% that of the article. Table 3 presents the difference in the average sentence length of the articles and the extracted summaries. Unlike in research paper summarizing, the extracted sentences of articles are of comparable length to the rest of the content.

Table 4.3: Average statistics of sentence count and word count in articles and the summaries

	Article	**Summary using KL**	**Summary using POS**
Total sentence count	420	19	62
Average sentence count	42.0	6.3	7.8
Total word count	10001	607	2231
Average word count	1250.1	202.3	278.9
Average sentence length	29.76	31.95	35.98

4.4.2 Readability test

The gunning fog readability test is performed on the research papers, abstracts, and summaries generated by both techniques separately. Figure 4.3 shows the results obtained. The Gunning fog index indicates the number of formal education required by the reader to understand the text. It typically ranges from 6 to 17. Any value greater than 17 is difficult to apprehend. On average, the readability of the research papers is 15.5, indicating it can be understood by a college junior (showcased in Table 3.7). The readability of the abstracts is less compared to the research paper except for R2 and R4. On average, the readability of the abstracts is 17.7. This minor increase compared to research papers can be justified by the fact that abstracts are typically produced for experts by specialists and require extensive background knowledge (Benbrahim and Ahmad (1995)).

The readability index of the summaries is higher than the abstract (22.9 on average for the external keyword list method and 22.2 on average for the POS tagging method). This difference is explained based on the index calculation. The Gunning Fog index is proportional to the average sentence length and the number of complex terms present in the text. As we saw in the previous section, the average sentence length of the summaries is high compared to the original text. Also, the research papers are lengthier and would include many closed-class words like ‘the’, ‘and’, ‘as’, etc. that would decrease the relative

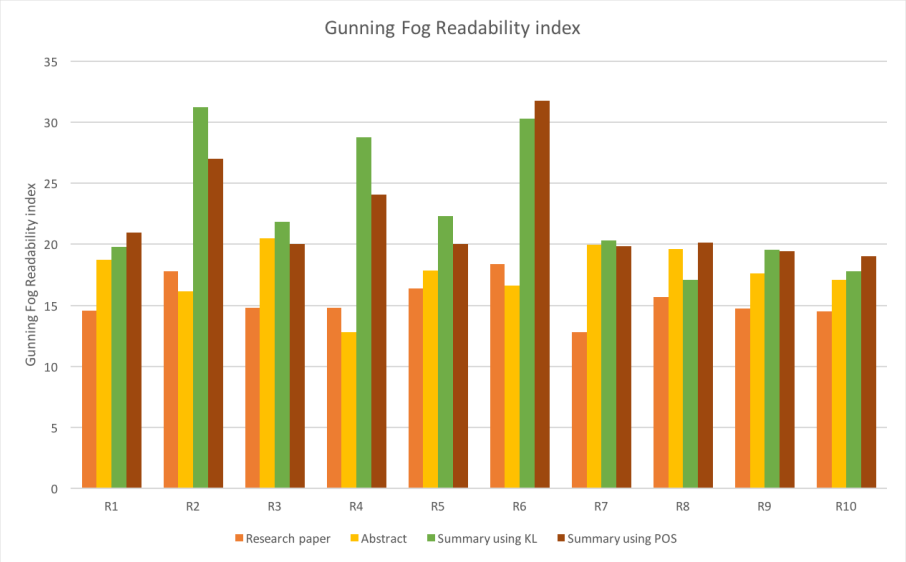

Figure 4.3: The Gunning Fog index of the research papers, their abstracts, and the summaries generated by the external keyword list and the POS tagging methods

Figure 4.4: The Gunning Fog index of the articles, and the summaries generated by the external keyword list and the POS tagging methods

number of complex words in the text. Therefore, the summaries are less readable than research papers.

The readability scores of the summaries of research papers R2, R4, and R6 have huge differences (double the score) from the abstract. The sentence count of the summaries of these papers and the abstract is almost the same, but the word count of the summaries is significantly high. It further confirms that the summaries have longer sentences, increasing the readability index value.

The readability scores of the science magazine articles and the summaries are presented in Figure 4.4. The average score for the articles is 18.6, 22.2 for the summaries using the external keyword list, and 18.1 for the summaries using the POS tagging approach. The readability of the articles is high because the articles are not as long as the research paper to have more closed-class words. Hence the articles have more complex words and are less readable compared to the research papers. The summaries produced are as readable as the articles. The reason is that the average sentence count of the articles and the summaries are almost the same as we saw earlier (Section 4.4.1).

4.4.3 Sentiment similarity

Sentiment analysis helps to understand the positivity or negativity in the text. In the context of evaluating the summaries, if both the input text and the summaries have matching emotion scores, it confirms that they transmit the information as intended and have not been distorted. VADER (Valence Aware Dictionary for Sentiment Reasoning) tool is used to evaluate the sentiment scores of the summaries. It returns a positive, negative, neutral, and compound score that is the normalized sum of the other three. For easy comparison, the compound score alone is adequate. The compound score ranges from -1 to 1, with scores close to -1, indicating more negativity, and scores close to 1, indicating more positivity.

Table 4.4 presents the compound scores obtained for the research papers, the abstracts, and the summaries generated using the external keyword list method and the POS tagging method. Except for one research paper (R6), all others have matching sentiment scores for the summaries. The research paper and summary both have negative content, but the summary contains more positive material than the research paper, that the compound score is positive. Considering 90% of the documents had matching compound ratings for both the original text and the summaries, it is reasonable to confirm that the information is presented accurately.

Table 4.5 presents the compound scores obtained for the articles and the summaries. The scores are matching for the summaries generated using the POS tagging method. But

Table 4.4: The compound scores obtained using VADER tool for the research papers, abstracts and summaries

Text	Sentiment score (compound)			
	Research paper	**Abstract**	**Summary using KL**	**Summary using POS**
R1	0.999	0.97	0.982	0.995
R2	0.995	0.977	0.861	0.989
R3	0.999	0.051	0.985	0.967
R4	0.992	0.714	0.806	0.841
R5	-0.998	-0.784	-0.969	-0.98
R6	-0.978	-0.959	-0.417	0.53
R7	0.999	0.807	0.996	0.876
R8	0.995	0.636	0.318	0.97
R9	0.996	0.636	0.978	0.988
R10	0.999	-0.735	0.986	0.87

Table 4.5: The compound scores obtained using VADER tool for the articles and summaries

Text	Sentiment score (compound)		
	Article	**Summary using KL**	**Summary using POS**
A1	0.916	0.856	0.671
A2	0.998	-	0.889
A3	0.977	-0.167	0.899
A4	-0.808	-	-0.995
A5	-0.914	-0.734	-0.91
A6	-	-	-
A7	-	-	-
A8	0.989	-	0.686
A9	0.99	-	0.381
A10	0.994	-	0.557

for the external keyword list method, the summary of A3 does not have matching values with the article score. As this summary did not capture as much neutral content as the original text, the normalized score moved closer to negative. On average, we could claim that the summaries capture the same amount of positive, negative, and neutral content as the original text.

4.4.4 Cosine similarity

Cosine similarity is one of the measures used to compare the syntactic similarity between the texts. It represents each document compared in a 2D space and measures the angle

between them. The cosine of this angle gives the similarity value. If the two documents are similar, their orientation will be the same, and the angle between them will be zero. It would result in a cosine value of one, indicating that the documents are identical. If the texts are oriented orthogonally, the similarity measure would be 0 (cosine of 90° is 0). A list of words with their counts is created to represent the text in the 2D space. Instead of including all the words in the document, the keywords (all nouns) are filtered in the process. Setting a threshold value to confirm the syntactic similarity between two input texts is depended on the user. Instead, the cosine similarity between the research papers and the abstracts is considered a guideline measure to compare the syntactic similarity between the generated summaries and the research papers.

Table 4.6: Cosine similarity of research papers with the abstract, and with the summaries generated using the external keyword list and the POS tagging method

Text	**Cosine similarity**		
	Research Paper/ Abstract	**Research Paper/ Summary using KL**	**Research Paper/ Summary using POS**
R1	0.357	0.524	0.515
R2	0.394	0.579	0.642
R3	0.198	0.451	0.457
R4	0.36	0.589	0.624
R5	0.289	0.669	0.596
R6	0.393	0.677	0.666
R7	0.182	0.485	0.422
R8	0.263	0.333	0.58
R9	0.334	0.542	0.494
R10	0.283	0.436	0.469
Total	3	5	5
Average	**0.3**	**0.5**	**0.5**
Standard Deviation	0.1	0.1	0.1

Table 4.6 showcases the cosine similarity measures between the research paper and the abstract, the research paper and the summaries produced using the external keyword method, and the research paper and the summaries generated using the POS tagging method. The average cosine similarity of the research papers and the abstracts is 0.3, indicating 30% similarity. But the similarity between the research paper and the summaries produced by both methods is 0.5 on average, showing 50% similarity. It shows that the generated summaries are syntactically more similar to the research paper than the human-generated summaries (abstracts).

Table 4.7 presents the results of the cosine similarity between the articles and the

generated summaries. Since articles do not have an abstract, there is no guideline to compare this measure. Similar to the research papers, the summaries generated have a similarity of 50% to the original article.

Table 4.7: Cosine similarity of the articles with the summaries generated using the external keyword list and the POS tagging method

Text	**Cosine similarity**	
	Article/ Summary using KL	**Article/ Summary using POS**
A1	0.569	0.62
A2	-	0.523
A3	0.202	0.554
A4	-	0.524
A5	0.65	0.624
A6	-	-
A7	-	-
A8	-	0.421
A9	-	0.484
A10	-	0.639
Total	1	4
Average	**0.5**	**0.5**
Standard Deviation	0.2	0.1

4.4.5 Jaccard similarity

Jaccard similarity index compares two documents based on the unique words present in them. If both the documents have the exact word list, then the similarity will be 1. The nouns in both documents are filtered out as two sets. The number of words present in both texts divided by the number of distinct words present in either of the texts gives the Jaccard index value. The main issue of using the Jaccard index to find the syntactic similarity of two documents is that it depends on the size of the document. The larger document would have more unique words than the smaller document, affecting the ratio. However, the purpose of utilizing this metric is to examine how much key information was captured by the summaries, compared to the abstracts.

Table 4.8 presents the Jaccard similarity between the research paper and the article (considered the guideline measure) and the research paper and the summaries generated using both methods. The average Jaccard similarity of the research papers and the abstracts is 0.1, indicating 10% similarity. The Jaccard similarity between the research papers and the summaries from both methods is 0.3 on average (30% similarity). It

Table 4.8: Jaccard similarity of research papers with the abstract, and with the summaries generated using the external keyword list and the POS tagging method

Text	**Jaccard similarity**		
	Research Paper/ Abstract	**Research Paper/ Summary using KL**	**Research Paper/ Summary using POS**
R1	0.139	0.274	0.265
R2	0.17	0.336	0.413
R3	0.05	0.203	0.209
R4	0.148	0.347	0.39
R5	0.092	0.448	0.355
R6	0.171	0.459	0.443
R7	0.042	0.236	0.178
R8	0.081	0.111	0.337
R9	0.119	0.293	0.244
R10	0.089	0.19	0.22
Total	1	3	3
Average	**0.1**	**0.3**	**0.3**
Standard Deviation	0.0	0.1	0.1

Table 4.9: Jaccard similarity of the articles with the summaries generated using the external keyword list and the POS tagging method

Text	**Jaccard similarity**	
	Article/ Summary using KL	**Article/ Summary using POS**
A1	0.324	0.385
A2	-	0.274
A3	0.041	0.307
A4	-	0.275
A5	0.422	0.39
A6	-	-
A7	-	-
A8	-	0.177
A9	-	0.234
A10	-	0.408
Total	1	2
Average	**0.3**	**0.3**
Standard Deviation	0.2	0.1

showcases that the summaries have captured more key information (only keywords are selected) than the abstracts. However, given that the increase is minor, it might be

because the overall word count of summaries is more than that of abstracts.

Similar to the case of research papers, the Jaccard similarity between the articles and the summaries produced by both methods is 0.3 (30% similarity). Table 4.9 presents the results for the ten articles summarized. As with cosine similarity, the articles lack abstracts (human-generated summary reference) to compare them.

4.4.6 Pearson correlation coefficient of relative word frequency

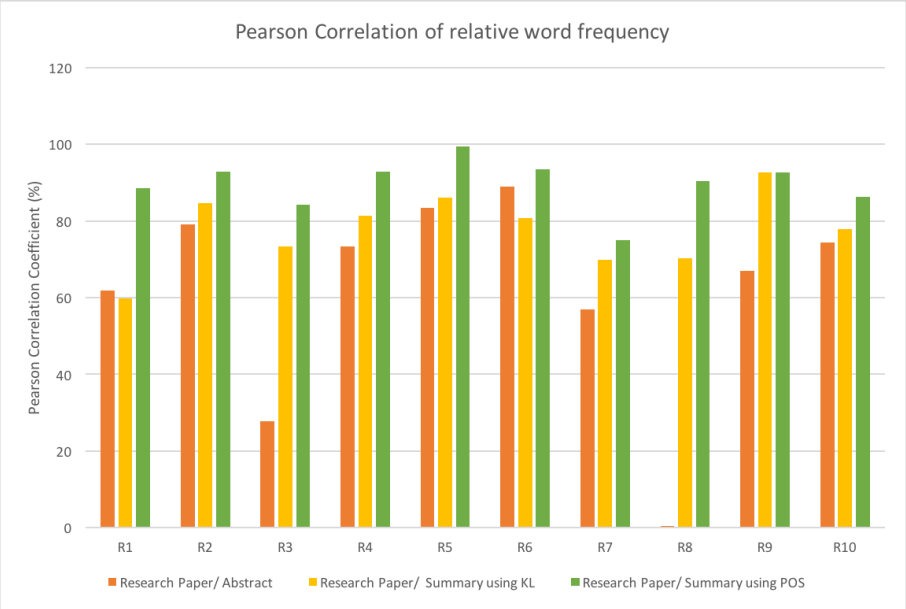

Figure 4.5: The Pearson correlation of relative frequencies of the research papers with the abstract, and with the summaries generated using the external keyword list and the POS tagging method

This evaluation technique helps to assess the lexical similarity between the original text and the summary. The hypothesis here is that if the word distribution of the input text is mirrored by the summary, we will have a good summary. The correlation between the word distribution of keywords in the research paper and the abstract is used as a reference and is compared to the correlation between the word distribution of the research paper and the summary. The relative frequencies of the keywords in the documents are used to determine the word distribution. The keywords include all the nouns in the text. The words' relative frequency is listed, and the Pearson correlation coefficient between the frequencies provides the similarity value. The correlation coefficient is a value ranging from -1 to 1, with -1 indicating a strong negative correlation and 1 indicating a strong positive correlation. Multiplying this value by 100 gives us a percentage value for easier comparison.

Figure 4.5 showcases the results obtained for the Pearson correlation of the word frequencies between the research papers and the abstracts and between the research papers and the summaries. The relative frequency of the keywords between the research papers and the abstracts match only by 61% on average. The summaries have better matching with 77.7% for the keyword extraction method and 89.6% for the POS tagging method. The higher values indicate that the summaries have managed to include the key terms with exact distribution.

Table 4.10: The Pearson correlation of relative frequencies of the articles and summaries

Text	**Article/ Summary using KL**	**Article/ Summary using POS**
A1	86.9	94
A2	-	89.2
A3	71.8	80.1
A4	-	78.1
A5	87.4	92.4
A6	-	-
A7	-	-
A8	-	76.8
A9	-	61.4
A10	-	82
Total	246.1	654
Average	**82.03**	**81.75**

Table 4.10 presents the Pearson correlation coefficient results for the articles and the generated summaries. Both the summarization methods have an average of 81% match. However, the absence of an abstract for the papers leaves us with no benchmark.

4.5 Discussion of results

The summary evaluation was implemented based on the word count, sentence count, readability measure, sentiment similarity, and syntactic similarity measures like cosine similarity, Jaccard similarity, and word distribution. A small dataset consisting of 10 science magazine articles and the referenced research papers were used for the experimentation. Comparison of the sentence count and the word count displayed the average length of sentences was high for the summaries compared to the original text. It demonstrates that the summarization algorithm extracts the longer sentences from the input text. It seems logical if we consider the necessity of the sentences to have more than three links to be bonded and labeled important (more than three repetitive keywords mean sentences have to be longer). The higher value for average sentence length in summaries accounts

for the higher readability score index. The summaries are hence not easily apprehensible. The sentiment analysis of the summaries and original texts gave similar scores indicating the summaries could maintain the emotion in the input text. As for the syntactic similarity, the cosine similarity, Jaccard similarity, and word distribution between the research papers and the summaries have better scores compared to the scores between the research papers and the abstracts. However, Jaccard similarity is inappropriate for assessing the summaries produced by extractive summarization algorithms because it compares the unique words in both documents to the entire set of words. Because extracted summaries comprise the sentences from the input text, all the words in the summary are also present in the input text. Nevertheless, this metric was retained for subsequent works on this research subject (especially for evaluating abstractive summaries).

Overall, the result of evaluation metrics looks promising to conclude that the text summarization using the lexical cohesion method is apt for summarizing research papers. It's worth noting that this summarization technique is more suited to summarizing research papers (long texts with good cohesiveness) than short pieces of text. However, the evaluation methodologies included here do not cover all aspects necessary to label a summary as good. An efficient summary must exclude all the unnecessary information in the input text. None of the metrics used here checks this criterion of evaluation.

Chapter 5

Conclusions & Future Work

5.1 Conclusion

The rising issue of the truth getting misrepresented due to the power of social media was the motivation for this dissertation work. A system that could effectively summarize the scientific papers was developed following the theory of lexical cohesion. The objective was to create a system that could capture the essential information from the research articles and use them as a guideline for the science magazine articles (next in the text-cline). Lexical cohesion holds the text together as a whole, and hence utilizing this for summarization resulted in good summaries. The system can summarize a text without any further training or dataset requirements. The algorithm counts the links between the sentences based on repetitions of keywords and labels the sentences as bonded if there are more than three links. The keywords are extracted using two methods: an external keyword list comprising the most commonly used terms in the context of the input text, a POS tagger identifying all nouns. It then classifies the highly-bonded sentences into topic-opening, topic-closing, middle, and marginal. The final summary extracts sentences from the first three classes to ensure continuity.

Apart from implementing an intelligent text summarizer, this work also showcased few techniques to evaluate the summaries automatically. The abstracts of the research papers (human-written summaries) are taken as a reference to verify the methods used. The evaluation included a statistical analysis of ten research papers and ten articles based on average sentence length, readability, syntactic similarity, and lexical similarity of the summaries to the input text. The generated summaries were less legible than the input material due to the length of the sentences and a drop in the closed class words, which typically boost the text's reading ease. In terms of syntactic and lexical similarity, the summaries outperformed the abstracts.

Although this is a domain-specific summarizing method, the methodology can be used in other domains simply by modifying the keyword list. Also, Lexical Cohesion theory to summarize the texts can be used for other languages texts. One of the most significant insights gained from experimenting with this summarizing method is that it does not function on all text types. Compared to research papers (large text with a high degree of cohesiveness), the summary algorithm could not summarize all the magazine articles (short text with little cohesion). The external keyword list method, in particular, performed poorly on summarizing the articles. This conclusion was derived using a small dataset of ten studies, which were summarized and evaluated when this report was written. A comprehensive evaluation of the system using a larger dataset and comparison to other summarization methods is currently being researched and will be presented as a research article later.

On a final note, the implemented summarization system has offered a technique to eliminate text misrepresentation on the first level of text-cline. The summaries produced using the system can aid the reader in understanding the content of texts without abstracts. However, better procedures must be implemented at subsequent levels to minimize the distortion of information when it finally reaches the general public. Though the inspiration for this dissertation work was the alarming issue of misrepresentation, the summarizing and evaluation approaches used here can be applied to numerous use-cases.

5.2 Limitations

Automatic summarization following Hoey's method of lexical cohesion proved to be efficient in summarizing the research papers. However, generalizing the performance of a system by experimenting with a smaller dataset is not adequate. Furthermore, the external keyword list method is domain-specific, and the experiment is conducted only on the coronavirus dataset. As a result, the efficacy of this approach cannot be determined at this stage.

Of the four repetitions illustrated by Hoey, Simple Repetition, Complex Repetition, and Simple Paraphrase were implemented completely in this work. In the case of Complex Paraphrase, only one scenario - antonyms - was considered. Even Hoey has claimed that the second scenario of finding the triangular link (simple paraphrase + complex repetition) is hard to automate. The omission of this repetition may have caused the system to miss a few links, reducing the summary's efficiency.

The time complexity of the summarization algorithm is O(n^2), where n is the sentence count of the input text due to the creation of the link matrix and bond matrix. It means as the length of input text increases, the time taken for summarization increases

exponentially.

Because of the longer sentences, the reading level of the summaries was high. Even if the summary has all the necessary information, it is not helpful if the reader cannot follow it. The algorithm limits the number of sentences in the summaries to keep them concise. As a result, the system might omit some vital information in longer texts.

Various evaluation metrics were introduced to test the method's efficacy. These metrics do not cover the overall aspects of labeling a summary as effective. A good summary must exclude the irrelevant information in the input text. None of the metrics presented here validates this. Furthermore, evaluating the summaries using the abstracts as a reference raises the question: whether the abstracts can be trusted as effective summaries?

5.3 Future Work

As pointed out in the limitations, if a system that could include the second scenario of the complex paraphrase is developed, it could create more efficient summaries. Such a method should also ensure that including Complex Paraphrase for linking sentences contributes significantly to the summary creation rather than increasing the time complexity. Furthermore, combining syntactic cohesion with the concept of lexical cohesion may or may not improve the efficiency of the summary. If proven effective, such a system can create meaningful summaries, or at the very least, emphasize the efficiency of employing lexical cohesion alone for the process.

Lexical cohesion is not unique to the English language and can effectively summarize texts in other languages. However, applying the method used in this work to other language texts is difficult. Because, unlike in the case of English, there is no WordNet dictionary to obtain synonyms or antonyms, or a pattern library to get singular or plural terms or even an external keyword list of words commonly used in the domain. Therefore, future work on implementing the method in other languages is beneficial.

A thorough evaluation of summaries with a larger dataset is essential to validate the method using the external keyword list. It is a work in progress and soon to be published (Jacob and Ahmad (2021)).

The automatic evaluation techniques presented here can validate any summarization system. Adding more methods like semantic similarity comparison would be outstanding future work. Also, a system that could automatically identify if the summary included any unnecessary information would improve the overall evaluation procedure.

Bibliography

Anthony, L. (2011). Antconc : A learner and classroom friendly, multi-platform corpus analysis toolkit.

Barzilay, R. and Elhadad, M. (2000). Using lexical chains for text summarization. *Advances in Automatic Text Summarization*.

Benbrahim, M. and Ahmad, K. (1995). Text summarisation: The role of lexical cohesion analysis. *The New Review of Document and Text Management*, 1.

Cerban, M. (2010). Lexical cohesion: Aspects of collocation using halliday and hasan's systemic model of cohesion. *Translation Studies: Retrospective and Prospective Views*.

Chauhan, K. (2018). Unsupervised text summarization using sentence embeddings. `https://medium.com/jatana/unsupervised-text-summarization-using-sentence-embeddings-adb15ce83db1`. Accessed on : 2021-08-08.

De Beaugrande, R. and Dressler, W. U. (1981). *Introduction to text linguistics / Robert-Alain de Beaugrande, Wolfgang Ulrich Dressler*. Longman London ; New York.

de Oliveira, P. C. F. (2005). How to evaluate the 'goodness' of summaries automatically. *PhD Thesis, University of Surrey*.

de Oliveira, P. C. F., Ahmad, K., and Gillam, L. (2002). A financial news summarization system based on lexical cohesion. In *In Proceedings of the International Conference on Terminology and Knowledge Engineering, Nancy, France.*

Ercan, G. and Cicekli, I. (2008). Lexical cohesion based topic modeling for summarization.

Gunning, R. (1952). *The technique of clear writing*. McGraw-Hill.

Halliday, M., Hasan, R., Halliday, R., Longman, P., and Quirk, R. (1976). *Cohesion in English*. A Longman paperback. Longman.

Harte, E. (2021). A fight against an infodemic. *Final year project, Trinity College Dublin*.

Hasan, R. and Flood, J. (1984). Understanding reading comprehension: Cognition, language, and the structure of prose.

Hoey, M. (1991). Patterns of lexis in text.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*.

Jacob, S. and Ahmad, K. (2021). Intelligent text summarization using lexical cohesion to reduce distortion of information. [To be published].

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., and Suchomel, V. (2014). The sketch engine: ten years on. *Lexicography*, pages 7–36.

Kulkarni, A. and Apte, S. (2014). An automatic text summarization using lexical cohesion and correlation of sentences. *International Journal of Research in Engineering and Technology*.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lloret, E., Plaza, L., and Aker, A. (2018). The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52:101–148.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2:159–165.

Mani, I. (2001). *Automatic Summarization*, volume 3.

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48.

Mosseri, A. (2019). Addressing hoaxes and fake news. `https://newsroom.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/`. Accessed on : 2021-08-02.

Nawaz, S. (2014). Cohesive ties and meaning comprehension.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation.

Radev, D. R. and Tam, D. (2003). Summarization evaluation using relative utility. New York, NY, USA. Association for Computing Machinery.

Saseedran, A. T. (2019). Intelligent summarization: Leveraging cohesion in text. *MSc. Dissertation, Trinity College Dublin*.

Silber, H. G. and McCoy, K. F. (2000). Efficient text summarization using lexical chains. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, page 252–255, New York, NY, USA. Association for Computing Machinery.

Spring, M. (2020). Coronavirus: The seven types of people who start and spread viral misinformation. `https://www.bbc.com/news/blogs-trending-52474347/`. Accessed on : 2021-08-08.

Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A., and Setiadi, D. R. I. M. (2020). Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*.

Zhang, X. and Ghorbani, A. (2019). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57.

Appendix

Dataset

Articles

1. https://www.sciencemag.org/news/2021/06/what-went-wrong-curevac-s-highly-anticipated-new-mrna-vaccine-covid-19 (Accessed on: Jun 27)
2. https://www.sciencemag.org/news/2021/02/world-s-largest-covid-19-drug-trial-identifies-second-compound-cuts-risk-death (Accessed on: Jul 12)
3. https://www.the-scientist.com/news-opinion/researchers-create-completely-self-contained-covid-test-with-3d-printed-parts-69080 (Accessed on: Aug 16)
4. https://www.sciencemag.org/news/2021/07/too-good-be-true-doubts-swirl-around-trial-saw-77-reduction-covid-19-mortality (Accessed on: Jul 31)
5. https://www.the-scientist.com/news-opinion/covid-19-vaccines-work-in-people-with-cancer-study-68930 (Accessed on: Jul 31)
6. https://www.the-scientist.com/news-opinion/covid-19-vaccines-appear-safe-during-pregnancy-early-data-68702 (Accessed on: Aug 16)
7. https://www.the-scientist.com/sponsored-article/glucometers-detect-sars-cov-2-infection-within-an-hour-68883 (Accessed on: Aug 7)
8. https://www.the-scientist.com/news-opinion/researchers-create-pathogen-sensing-face-mask-68945 (Accessed on: Aug 7)
9. https://www.the-scientist.com/news-opinion/spike-protein-deletions-linked-to-covid-19-surges-preprint-68892 (Accessed on: Aug 8)
10. https://www.the-scientist.com/news-opinion/sars-cov-2-antigens-leaking-from-gut-to-blood-might-trigger-mis-c-68845 (Accessed on: Aug 8)

Research Papers

1. Kremsner, P., Mann, P., Bosch, J., Fendel, R., Gabor, J. J., Kreidenweiss, A., Kroidl, A.,Leroux-Roels, I., Leroux-Roels, G., Schindler, C., Schunk, M., Velavan, T. P., Fotin-Mleczek, M., M uller, S., Quintini, G., Sch onborn-Kellenberger, O., Vahrenhorst, D.,Verstraeten, T., Walz, L., Wolz, O.-O., and Oostvogels, L. (2020). Phase 1 assessment of the safety and immunogenicity of an mrna-lipid nanoparticle vaccine candidate against sars-cov-2 in human volunteers.

2. Group, R. C., Horby, P. W., Pessoa-Amorim, G., Peto, L., Brightling, C. E., Sarkar, R.,Thomas, K., Jeebun, V., Ashish, A., Tully, R., Chadwick, D., Sharafat, M., Stewart,R., Rudran, B., Baillie, J. K., Buch, M. H., Chappell, L. C., Day, J. N., Furst, S. N.,Jaki, T., Jeffery, K., Juszczak, E., Lim, W. S., Montgomery, A., Mumford, A., Rowan,K., Thwaites, G., Mafham, M., Haynes, R., and Landray, M. J. (2021). Tocilizumab in patients admitted to hospital with covid-19 (recovery): preliminary results of arandomised, controlled, open-label, platform trial

3. de Puig, H., Lee, R. A., Najjar, D., Tan, X., Soekensen, L. R., Angenent-Mari, N. M.,Donghia, N. M., Weckman, N. E., Ory, A., Ng, C. F., Nguyen, P. Q., Mao, A. S., Fer-rante, T. C., Lansberry, G., Sallum, H., Niemi, J., and Collins, J. J. (2021). Minimallyinstrumented sherlock (misherlock) for crispr-based point-of-care diagnosis of sars-cov-2and emerging variants. 7(32)

4. Cadegiani, F. A., do Nascimento Fonseca, D., McCoy, J., Zimerman, R. A., Mirza, F. N.,do Nascimento Correia, M., Barros, R. N., Onety, D. C., Israel, K. C. P., de Almeida,B. G., Guerreiro, E. O., Medeiros, J. E. M., Nicolau, R. N., Nicolau, L. F. M., Cunha,R. X., Barroco, M. F. R., da Silva, P. S., de Souza Ferreira, G., da Costa Alcˆantara,F. R. P., Ribeiro,ˆA. M., de Almeida, F. O., de Souza Silva, A. A., do Rosario, S. S.,de Souza Paulain, R. W., Reis, A., Li, M., Thompson, C. E., Nau, G. J., Wambier,C. G., and Goren, A. (2021). Efficacy of proxalutamide in hospitalized covid-19 patients:A randomized, double-blind, placebo-controlled, parallel-design clinical trial.

5. Thakkar, A., Gonzalez-Lugo, J. D., Goradia, N., Gali, R., Shapiro, L. C., Pradhan, K.,Rahman, S., Kim, S. Y., Ko, B., Sica, R. A., Kornblum, N., Bachier-Rodriguez, L.,McCort, M., Goel, S., Perez-Soler, R., Packer, S., Sparano, J., Gartrell, B., Makower,D., Goldstein, Y. D., Wolgast, L., Verma, A., and Halmos, B. (2021). Seroconversionrates following covid-19 vaccination among patients with cancer

6. Shimabukuro, T., Kim, S., Myers, T., Moro, P., Oduyebo, T., Panagiotakopoulos,

L.,Marquez, P., Olson, C., Liu, R., Chang, K., Ellington, S., Burkel, V., Smoots, A.,Green, C., Licata, C., Zhang, B., Alimchandani, M., Mba-Jonas, A., Martin, S., andMeaney-Delman, D. (2021). Preliminary findings of mrna covid-19 vaccine safety inpregnant persons.New England Journal of Medicine, 384.

7. Singh, N. K., Ray, P., Carlin, A. F., Magallanes, C., Morgan, S. C., Laurent, L. C.,Aronoff-Spencer, E. S., and Hall, D. A. (2021). Hitting the diagnostic sweet spot: Point-of-care sars-cov-2 salivary antigen testing with an off-the-shelf glucometer.Biosensorsand Bioelectronics, 180:113111

8. Nguyen, P.Q., Soenksen, L.R., Donghia, N.M. et al. Wearable materials with embedded synthetic biology sensors for biomolecule detection. Nat Biotechnol (2021). https://doi.org/10.1038/s41587-021-00950-3

9. Venkatakrishnan, A., Anand, P., Lenehan, P., Ghosh, P., Suratekar, R., Siroha, A.,Chowdhury, D. R., O'Horo, J. C., Yao, J. D., Pritt, B. S., Norgan, A., Hurt, R. T.,Badley, A. D., Halamka, J. D., and Soundararajan, V. (2021). Antigenic minimal-ism of sars-cov-2 is linked to surges in covid-19 community transmission and vaccinebreakthrough infections

10. Yonker, L. M., Gilboa, T., Ogata, A. F., Senussi, Y., Lazarovits, R., Boribong, B. P.,Bartsch, Y. C., Loiselle, M., Rivas, M. N., Porritt, R. A., Lima, R., Davis, J. P.,Farkas, E. J., Burns, M. D., Young, N., Mahajan, V. S., Hajizadeh, S., Lopez, X. I. H.,Kreuzer, J., Morris, R., Martinez, E. E., Han, I., Jr., K. G., Barry, N. C., Thompson,D. B., Church, G., Edlow, A. G., Haas, W., Pillai, S., Arditi, M., Alter, G., Walt, D. R.,and Fasano, A. (2021). Multisystem inflammatory syndrome in children is driven byzonulin-dependent loss of gut mucosal barrier.The Journal of Clinical Investigation,131