# Assignment 3: NLP Application Mini Project Summarization Application

Sinuo Wang - a1713814, Duc Nhan Do - a1848777

The University of Adelaide

## 1 Executive Summary

This report describes the extension of Assignment 2 which centers around constructing an information retrieval system that uses TF-IDF and GloVe [13] in conjunction with a knowledge base to retrieve relevant articles based on user input queries. The extended system generates output answers by summarizing the top 5 matching articles. Two methods were employed to build the summarizer: **(1)** Extractive summarizer, which utilizes the Weighted Sentence approach; and **(2)** Abstractive summarizer that employs the BERT2BERT [4] architecture to generate the summary outputs. The performance of each approach are both evaluated based on the ROUGE scores and human evaluations. This report provides a detailed explanation of both methods, a comparison of the results, and presents the findings.

## 2 Methods

*Summarization* is a natural language processing task that involves condensing a larger text into a shorter version while retaining the essential information and preserving the overall meaning of the original text. The primary objective of summarization is to facilitate readers in efficiently comprehending the central points of a document without the need to read the entire text. Such systems can significantly reduce the time and effort required by users to comprehend the crucial points of voluminous texts. Summarization can be approached in various ways, but with two main categories being **Extractive Summarization** and **Abstractive Summarization** [7].

– *Extractive Summarization*: These methods rely on extracting the most relevant parts, such as phrases and sentences, from a text and stacking them together to create a summary. Therefore, identifying the proper sentences for summarization is paramount in an extractive method.

– *Abstractive Summarization*: Such approach involves employing advanced natural language processing (NLP) techniques to generate a new summary that may not contain the same phrases or sentences as the original text. As it requires the ability to understand the content, generate fluent language, and balance relevance and brevity. Machine learning algorithms, such as deep learning models, have been successful in natural language understanding, generation, and balancing trade-offs, making them essential for high-quality abstractive summarization.

In this project, we explore and examine both summarization categories: Weighted Sentence and TextRank utilizing NLTK library for the extractive method and the BERT2BERT models for the abstractive approach.

### 2.1 Weighted Sentence and TextRank – Extractive Summarization

This approach is an extractive summarization technique using **NLTK** [1] to extract the most relevant sentences from the given text. NLTK stands for *Natural Language Toolkit*. It is a leading platform for building Python programs to work with human language data. It provides a set of tools and resources that can be used to perform various natural language processing tasks, such as tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and machine translation [6]. It also provides a comprehensive suite of libraries and tools that can be used for data preprocessing, feature extraction, and model training and evaluation. Therefore, NLTK is an excellent tool for building a wide range of natural language processing applications quickly and efficiently.

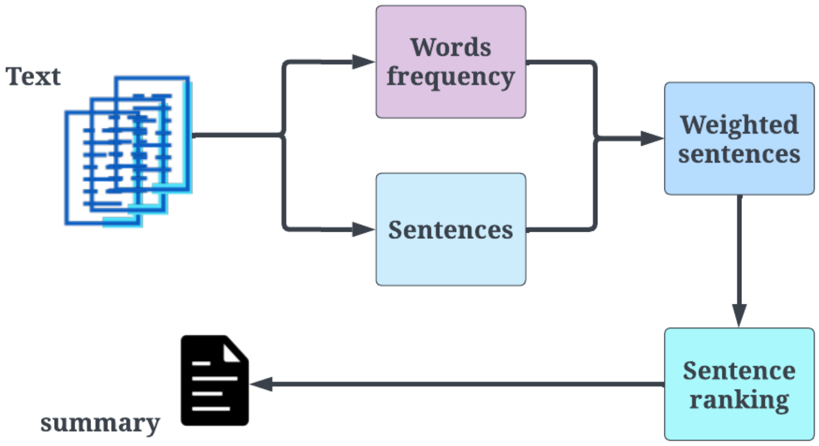


Fig. 1: The workflow of weighted sentences algorithm.

This NLTK method applies two approaches to extract relevant sentences and form a summary: **weighted sentence** and **TextRank** algorithm. *The weighted sentence* technique is a popular approach used in automatic summarization tasks. As shown in Fig. 1, this technique involves assigning weights to each sentence in a given text based on its importance to the overall meaning of the document. The weight of a sentence in this assignment is determined by analyzing the frequency of words after tokenizing the text. Once the weights are assigned, a predefined number of sentences with the highest weights are selected and combined to create a summary of the original document. This approach ensures that the most important sentences are included in the summary while less crucial or redundant sentences are excluded [11].

*TextRank* is another popular automatic summarisation technique, particularly in extractive summarization. This approach is based on the PageRank algorithm used by Google to rank web pages. Text rank represents the document as a graph, where each sentence is a node, and the edges between nodes represent the similarity between sentences. The similarity between sentences in this assignment is determined using cosine similarity after converting each sentence into a vector. Once

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

the graph is created, the PageRank algorithm is applied to the graph to determine each sentence's importance. The sentences with the highest scores are then selected and combined to summarize the original document [7]. The TextRank approach can effectively generate summaries that capture the most important information in the original document while reducing redundancy and eliminating less important information. The workflow of the TextRank algorithm is described in Fig. 2

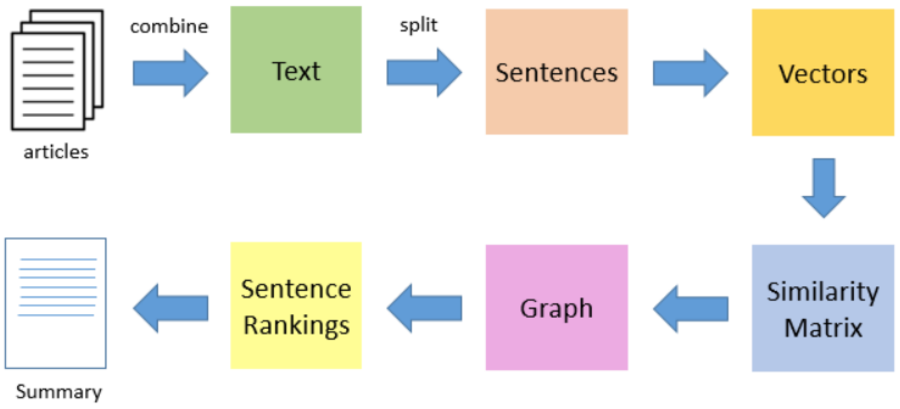


Fig. 2: The workflow of TextRank algorithm [7].

### 2.2 BERT2BERT – Abstractive Summariztion

Despite its cost-effectiveness and ease of deployment, the above-mentioned method exhibits limitations in adequately capturing semantic meaning and coherence between sentences, leading to the generation of unnatural summaries for human consumption. To address these shortcomings, we propose leveraging the state-of-the-art pre-trained language model, BERT (Bidirectional Encoder Representations from Transformers) [5] to enhance content understanding and improve the overall coherence, thus facilitating better comprehension and readability for human users. BERT [5] is a pre-trained language model that employs self-attention to encode the interrelationships among various tokens in a sequence. This mechanism enables the model to grasp intricate dependencies between different sections of the text, resulting in the creation of contextualized embeddings of the input sentence. Furthermore, BERT is pre-trained over a massive corpus of unlabelled text data in a self-supervised manner through Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). Specifically, for the task at hand, we employ pre-trained language models, SciBERT [1] [2] and COVID-SciBERT [3], to leverage their extensive knowledge of natural language and fine-tune them for the task of CORD-19 [14] scientific article summarization. SciBERT [1], which is a BERT model fine-tuned on a corpus of 1.14M scientific papers from Semantic Scholar. COVID-SciBERT [4], on the other hand, is a further fine-tuned SciBERT [1] model trained on the CORD-19 dataset [14]. In 2021, Chen et al. proposed a novel artecture, named BERT2BERT [4], which combines two BERT models: source model pre-trained on a large corpus of text data, and a target model

[2] https://huggingface.co/allenai/scibert_scivocab_cased
[3] https://huggingface.co/lordtt13/COVID-SciBERT
[4] https://huggingface.co/lordtt13/COVID-SciBERT

fine-tuned on a specific downstream task using the source model's weights as initialization, which can help the target model learn better representations of the language for the target task. In this project, we froze the weights of the source model and unfroze the weights in the target model for CORD-19 [14] summarization. This approach addresses the issue of most Large Language Models (LLM) being trained from scratch without leveraging existing pre-trained models, which can result in wasteful utilization of computational resources. Building on the previous work and the idea of transfer learning, we conducted an experiment with two BERT2BERT [4] pipelines. These pipelines employed a combination of two COVID-SciBERT [5] models and two SciBERT [1] models as the two BERT components. By utilizing these pre-trained models with the representation learning and transfer learning scheme in BERT2BERT [4], we aim to improve the summarization of COVID-19 [14] research articles.

## 3 Implementation

We have partitioned the entire implementation process into two phases, namely retrieval and summarization. In the first phase, the retrieval code from Assgin2 was repurposed to carry out the task of text retrieval. The best matching text is retrieved based on the posed question, and subsequently advanced to the next stage of the process. Thereafter, our trained model is applied to summarize the retrieved text and generate the answer to the question.

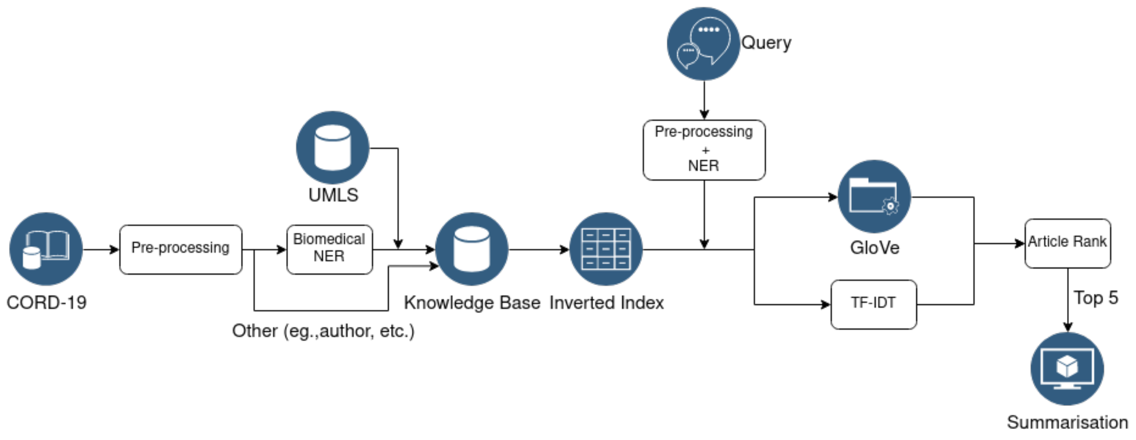


Fig. 3: System Structure

### 3.1 Document Retrieval

**Text Pre-processing:** As shown in Fig.3, the document retrieval pipeline consists of text pre-processing, knowledge base construction, article indexing, and text matching. Text pre-processing is an essential step before performing Named Entity Recognition (NER) on any given text. The purpose of text pre-processing is to clean, and transform the text data into a format that can be

[5] https://huggingface.co/lordtt13/COVID-SciBERT

effectively used for NER. In the present system, the pre-processing of text data involves the removal of in-text references and non-alphanumeric characters from the articles, followed by tokenization. It is worth noting that periods and underscores are preserved in the text since they serve critical roles in distinguishing sentences and identifying medical terms, such as 'SARS-CoV-2', respectively. Removal of these special characters would have a significant impact on the performance of Named Entity Recognition (NER) models. Hence, preserving these characters is necessary to maintain the accuracy of NER systems in recognizing and extracting entities from the text.

**Knowledge Base Construction:** During the construction of our knowledge base, we conducted a comprehensive scan of a large corpus of texts using a process pipeline that involved Named Entity Recognition (NER), abbreviation resolution, and entity linking to the Unified Medical Language System (UMLS) knowledge base [3]. Given the extensive size of the corpus, we limited our analysis to the title, abstract, and the initial and final 500 words of the body text, which were deemed to be indicative of the entire text. We utilized the 'en_core_sci_sm' pre-trained model [6] from ScispaCy [12] for Named Entity Recognition (NER). In addition to NER, we incorporated an extra pipeline called 'entity_linker' to link the recognized entities to the UMLS (Unified Medical Language System) knowledge base [3], which is comprised of approximately 3M biomedical concepts. The entity linker maps each recognized entity to a known entity or concept in the UMLS knowledge base, while simultaneously resolving any abbreviation issues. Our approach involves linking each recognized entity to the most relevant match in the UMLS knowledge base [3], with a pre-determined threshold value of 0.8. If a recognized UMLS entity is not the same as its canonical name or any of its stored aliases, this entity name is appended to the alias list and the UMLS canonical name is assigned as the key for this entity in the constructed knowledge base. As a result, text normalization is achieved through the use of the UMLS concept identifier (CUI), which maps each recognized entity to its corresponding CUI in the UMLS knowledge base, thereby ensuring consistency in the representation of biomedical concepts. The use of text normalization allows for the creation of a standardized and accurate representation of entities across the entire corpus of biomedical texts, facilitating robust and efficient information retrieval.

**Article Indexing:** The utilization of the information stored in the knowledge base is imperative for the efficient construction of the inverted index. Specifically, the article indexing method adopted is the inverted index by named entities. In this approach, each canonical name present in the knowledge base serves as a key in the inverted_index dictionary. The corresponding value associated with each key is a list of tuples, where each tuple comprises the ID of the paper that mentions the entity and the corresponding term frequency of the entity in that paper. It is important to note that the term frequency considered here is the value resulting from text normalization, as the term frequency in the knowledge base accounts for UMLS aliases and other names that map to the same UMLS concept. By utilizing the inverted index, the search space during document retrieval can be significantly reduced.

**Text Matching Utility – Retriever:** In the text match utility, the pre-trained embedding GloVe 840B 300d [13] [7] was utilized as a means of capturing the context of the word in the embedding. GloVe[13] embedding is pre-trained to capture the co-occurrence probability explicitly. The method aims to construct an embedding that is not solely based on the probability of the word itself, but also on the probability of its co-occurrence within a given context. The usage of the GloVe embedding

[6] https://allenai.github.io/scispacy/
[7] https://nlp.stanford.edu/projects/glove/

[13] in this involved downloading the GloVe file and opening it as a text file. A dictionary was then created, with the word serving as the key and its corresponding embedding vector as its value. Although the GloVe embedding [13] has demonstrated its usefulness in NLP tasks, its applicability to biomedical texts is limited because it was not extensively trained on medical terms. In this study, the GloVe embedding [13] was found to cover only 44.63% of UMLS concepts in our CORD-19 dataset. To mitigate this limitation, we incorporated the TF-IDF method as an additional measure of text ranking scores. While TF-IDF does not suffer from word coverage issues, it does not capture semantic meaning. To overcome these, both GloVe [13] and TF-IDF embeddings were used together to provide a more comprehensive approach to text retrieval, particularly in out-of-vocabulary scenarios.

In this information retrieval (IR) system, the query undergoes pre-processing by removing stopwords and is subject to named entity recognition (NER) using two models, namely, 'en_core_sci_sm' [8] for medical entities and 'en_core_web_sm' [9] for general entities. This is necessary since the SciSpaCy medical NER model is inadequate in recognizing general entities. The system matches entities and retrieves the posting list to reduce the paper search space. If the entity in the query is matched to a canonical name in the inverted index key, the system stores the entity for further embedding and TF-IDF computation and obtains the IDs of the articles mentioning the entity. If the entity in the query is an alias of a canonical name, its canonical name is obtained, and the same steps are repeated. The score for all documents in the reduced search space is computed. The retrieval rank score comprises two parts: the TF-IDF and the cosine similarity of the query and document vectors. The query and document are both represented by their recognized entities, $E_q$ and $E_d$. If an entity, $e \in E$, comprises multiple word tokens, it is represented as a mean-pooling of all the of the tokens $V_e = \frac{\sum_{w \in e} V_w}{|e|}$, where $V_w \in R^{300}$ is the GloVe word embedding for the word w in entity $e$. The query and document embedding vectors are mean-pooling vectors of all their recognized entities embeddings, which are represented as $V_q = \sum_{e \in E_q} V_e$ and $V_d = \sum_{e \in E_d} V_e$ respectively. Therefore, both the query embedding $V_q$ and document embedding $V_d$ are in the shape of $R^{300}$. The resulting GloVe cosine similarity is computed as: $\cos(\theta) = \dfrac{V_q \cdot V_d}{\|V_q\|_2 \|V_d\|_2}$. To overcome the limited medical vocabulary coverage of the GloVe embedding, both the TF-IDF and the GloVe embedding cosine similarity are used. The cosine similarity list and the TF-IDF list for all the candidate documents are standardized in the range of 0-1. The system averages the standardized TF-IDF score and the standardized embedding cosine similarity for each document, which is utilized as the final ranking score. This scoring scheme allows rare medical words not in the GloVe embedding to obtain some ranking scores from the TF-IDF part. Finally, the documents are ranked based on the GloVe + TF-IDF fused scores, and the first 5 documents are returned to the summarizer to perform the summarization task.

### 3.2 Summarization

**Extractive Summarizers:** NLTK is a natural language processing tool that offers a quick and facile method without necessitating any extensive training. As compared to deep learning models, which require an abundance of data and computational resources to train, the NLTK approach can be implemented promptly. Nonetheless, to effectively utilize the NLTK approach, certain preprocessing steps must be undertaken. For instance, when summarizing a lengthy article, the body text

[8] https://allenai.github.io/scispacy/
[9] https://spacy.io/models/en

can be split into smaller segments, typically consisting of 50 sentences, to facilitate the summarization technique. This approach enables a more streamlined and focused collection of information to be employed in the summarization process. Additionally, certain preprocessing measures, such as stopword removal and lowercase conversion, are executed to generate a sanitized text for the summarizer. These actions guarantee that the summarization process is more precise and germane to the provided query. The *weighted sentence* algorithm assigns weight to each sentence based on its relevance to the overall document. This weight can be achieved by calculating the frequency of occurrence of each word in the text, and then the weight is normalized by dividing it by the number of tokens in the sentence. Once the weights are assigned, the sentences can be ranked based on their importance. The *TextRank* algorithm utilizes graph theory to identify the most relevant sentences in the text. This algorithm creates a graph where each sentence is converted to a vector using the ***"glove.6B.100d"*** pre-trained word embedding model to represent a node, and the similarity between sentences is represented as edges between nodes. Finally, the most important sentences are identified by calculating each sentence's PageRank score in the graph and combined to generate a text summary.

**Abstractive Summarizers:** Given that the maximum sequence length of BERT is limited to 512 tokens, while the body text of articles typically ranges between 3000-4000 words in length, a dataset must be created to fulfill this requirement. We utilized the NLTK Weighted Sentence method to extract represntative sentences. This method allowed us to extract sentences that were of significant relevance to the article while ensuring that the total length of the extracted text remained within 500 words, which is a suitable length considering the use of the WordPiece tokenizer by BERT. The addition of the beginning-of-sentence (BOS) and end-of-sentence (EOS) tokens to indicate the start and end of sentences for both the encoder and decoder models, further justified the choice of 500 words as opposed to the input maximum of 512. To construct the dataset, we used the abstracts as the ground truth outputs and cropped them to 250 words. The resulting dataset was randomly split into training, validation, and test sets, with proportions of 0.7, 0.1, and 0.2, respectively. For decoder generation, we employed beam search with a beam size of 5 as an alternative to the conventional greedy search, as it has been shown to exhibit superior performance by considering multiple candidate sequences and expanding the search space. In contrast to greedy search, beam search selects the most promising sequence based on a scoring function that incorporates both the probability of the generated sequence and a length penalty. To address the issue of text repetition, we set the *no repeat ngram size* to 2, which prevented the two neighboring tokens from being the same. During training, a relatively small learning rate of 1e-4 was used to avoid flushing the pre-trained knowledge of the original model. The training and evaluation batch sizes were set to 4, taking into account the limited local computational power. We trained the model over the training set for 5 epochs using the Adam optimizer[9] with an epsilon value of 1e-8.

## 4 Results

In this project, the abstract of articles is used as the reference summary for evaluation. The metrics used for the evaluation combine the ROUGE score as an automatic evaluation to compare the methods' performance and human evaluation in the final output to check the semantic meanings of the generated answer.

### 4.1 Summary Evaluations

**Automatic Evaluation** ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a popular set of evaluation metrics used in NLP and text summarization research to measure the quality of system-generated summaries. The metric is based on the overlap of n-gram units (continuous sequences of n words) between the system-generated summary and the reference summaries. ROUGE includes several versions, such as **ROUGE-1, ROUGE-2**, and **ROUGE-L** [10].

- **ROUGE-1** measures the overlap of *unigram* (single-word) units by counting the number of shared *unigrams* in the output and reference summaries. Then, the result is divided by the total number of *unigrams* in the reference summary.
- **ROUGE-2** measures the overlap of *bigram* (two-word) units between the output and the reference summaries by counting the number of shared *bigrams* of both summaries and then dividing it by the total number of *bigrams* in the reference summary.
- **ROUGE-L** is a variant that measures the longest common subsequence of words in the output and reference summaries. First, the longest common subsequence is computed by finding the longest sequence of words that appear in both summaries. Then, the ROUGE-L score is computed by dividing the length of the longest common subsequence by the total number of words in the reference summary.

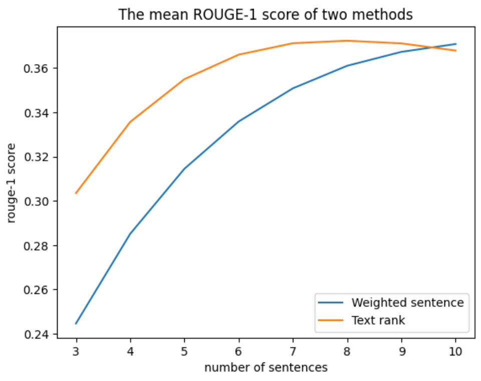
ROUGE scores range from 0 to 1, where 1 indicates a perfect match between the generated and reference summaries. In practice, ROUGE scores are typically used to compare the performance of different summarization systems, with higher scores indicating better performance.

| **Method** | ROUGE-1 | ROUGE-2 | ROUGE-L | Avg. |
|---|---|---|---|---|
| **NLTK (Weighted Sentence)** | 37.09 | 12.50 | 18.81 | 22.80 |

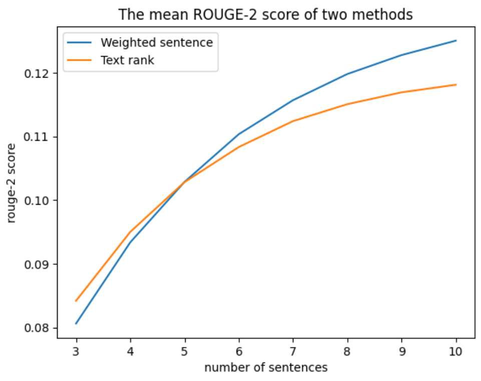
Table 1: The best score of NLTK on weighted sentence technique, which takes 10 sentences to form a summary.

**Weighted Sentence and TextRank** The NLTK approach is an extractive summary technique, so the number of extracted sentences to form a summary is the main factor affecting the generated summary score. Therefore, this approach is evaluated using the ROUGE score and analyzed in different summary lengths. Because the summary length increases, the output may include more information and details, making the evaluation process more robust and comprehensive. This assignment evaluated the effectiveness of two text summarization methods: the *weighted sentence* method and the *text rank* method, using ROUGE-1, ROUGE-2, and ROUGE-L. As shown in Figure. 4, the *weighted sentence* method had the highest score among the three metrics used in the evaluation. Interestingly, it was found that the more sentences included in the generated summary, the higher the score obtained by the *weighted sentence* method.

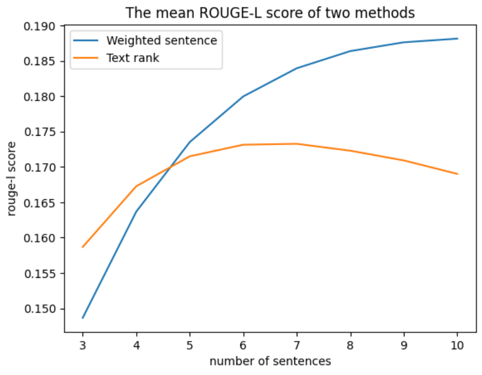
On the other hand, the *text rank* method showed mixed results when evaluated using the ROUGE-1, ROUGE-2, and ROUGE-L metrics. In ROUGE-1, the best score was obtained when the summary length was 7 sentences, and the score gradually decreased as the number of sentences increased. However, in ROUGE-2, the score positively correlated with the number of sentences in the output summary. Finally, in ROUGE-L, the score was initially higher than in ROUGE-2 but gradually fell as the number of sentences increased. Despite receiving relatively low scores, both the *weighted*



(a) ROUGE-1



(b) ROUGE-2



(c) ROUGE-L

Fig. 4: The ROUGE scores of NLTK approach with two methods: weighted sentence and text rank.

*sentence* and the *text rank* methods could generate fairly good summaries without requiring training and can be used directly for summarization. These methods could be a viable option when resources for deep learning models are lacking.

In general, the NLTK approach is characterized by its ease of deployment, no model training requirements, and grammatical accuracy. However, its performance is deemed suboptimal due to its reliance on frequency and PageRank algorithms, which are unable to fully capture the semantic meaning of the original text. Moreover, summaries produced by extracting sentences tend to lack coverage of the entire document and coherence between sentences. Therefore, we will explore on the state-of-the-art abstractive summarization methods to address these limitations, given their ability to contextualize and better comprehend the original text to generate concise and readable summaries that effectively capture the most significant information.

**BERT2BERT:** The evaluation of the BERT2BERT [4] models' performance, as presented in Table 2, demonstrates that both SciBERT [1] and COVID-SciBERT [11] perform comparably, with SciBERT [1] exhibiting slightly better results. Specifically, SciBERT [1] outperforms COVID-SciBERT in Rouge-2 score, while COVID-SciBERT marginally surpasses SciBERT in Rouge-1 score. This dif-

[11] https://huggingface.co/lordtt13/COVID-SciBERT

| **BERT Type** | ROUGE-1 | ROUGE-2 | ROUGE-L | Avg. |
|---|---|---|---|---|
| **COVID-SciBERT** [1] | 80.55 | 35.87 | 75.82 | 64.08 |
| **SciBERT** [10] | **80.38** | **36.13** | **75.82** | **64.11** |

Table 2: Comparison with BERT2BERT [4] Architecture with SciBERT [1] and COVID-SciBERT on the test set

ference in performance could potentially be attributed to the fact that COVID-SciBERT has a more comprehensive understanding of COVID-related terminologies, leading to larger unigram overlaps with the ground truth COVID summary. Notably, the observed similarity in the performance of the two models does not align with the initial expectation that COVID-SciBERT, with its increased pre-trained knowledge more relevant to the task at hand, would outperform SciBERT [1]. Further experiments, such as hyper-parameter tuning for all models and gathering additional high-quality summarization data, would be necessary to arrive at a more definitive conclusion. However, these initiatives are currently limited by computational resources and will be considered for future work.

Based on the obtained results, the Weighted Sentence and BERT2BERT with SciBERT [4,1] components were deemed as the effective summarizers and, therefore, selected for integration into our final retrieval summarization system.

### 4.2 Human Evaluation

The use of ROUGE as a standard metric for text summarization evaluation is widespread. However, it is important to acknowledge that this metric has certain limitations. Specifically, ROUGE only measures the degree of overlap between the generated and reference summaries in terms of n-gram units, while disregarding the semantic coherence between them. Furthermore, ROUGE assumes that the reference summaries are an absolute standard, which may not always be the case. In addition, the coherence and comprehensibility of the machine-generated summary are not accounted for by ROUGE. Therefore, it is recommended to complement ROUGE scores with human assessments to provide a more comprehensive evaluation of the summarization system. To this end, we performed human evaluations by displaying the retrieval output with the corresponding summary results of both summarizers of the pipeline, as shown in Table 3. Our analysis revealed that the Weighted Sentence method effectively captured the article's keyword (Query 1: **mRNA**; Query2: **lung function**), but lacked coverage of the article's content. Furthermore, we observed that extractive summarization suffered from a lack of coherence between the sentences. On the other hand, the BERT2BERT [4] summarizer was able to capture the key concepts of the articles, usually not the exact keyword in the query (Query 1: **mRNA** and **RNA**; Query 2: **lung function** and **respiratory**). However, the fluency and grammar of the generated summary were suboptimal, such as, sentences tends to stop suddently, and grammar mistakes. This is because the BERT2BERT [4] architecture combines two separate BERT models, where only the individual models are further pre-trained on scientific text for MLM and NSP tasks. Consequently, the entire encoder-decoder model only contains pre-trained knowledge about related topics but has never been trained on text generation tasks, resulting in a lack of ability in text generation and the delivery of unsatisfactory fluency and grammar.

In comparison to the retrieval results presented in Assignment 2, it is notable that readers can now rapidly and succinctly access the information they require without having to read through the entire articles. Nevertheless, the generated summaries do exhibit some areas for potential improvement.

| **What is the relation between mRNA and COVID?** | |
|---|---|
| 1st Title | Virus infection-induced host mRNA degradation and potential application of live cell imaging |
| Weighted Sentence (2) | Host RNA degradation also reported IAV-infected cells though sensitivity host mRNAs IAV-induced host shutoff varies. Intriguingly prokaryotic viruses bacteriophages also trigger rapid host mRNA degradation. |
| BERT2BERT [4] | the direct rna - dependent rna polymerase rdrp plays important role viral infection. it also plays |
| **Will covid cause lung function problems?** | |
| 1st Title | Early Childhood Pneumonia Is Associated with Reduced Lung Function and Asthma in First Nations Australian Children and Young Adults |
| Weighted Sentence (3) | We hypothesised early childhood pneumonia associated poorer lung function asthma age pneumonia first occurring modifies associations decreasing age pneumonia greater impact. A total 48 subjects verifiable pneumonia 27 subjects first pneumonia three years 13 three five years 9 five years age. Subjects whose pneumonia occurred age five years combined plotted subjects Figure 1 including whose pneumonia occurred age five years never-pneumonia status. |
| BERT2BERT [4] | background the coronavirus disease 2019 covid - 19 caused severe acute respiratory syndrome corona virus |

Table 3: Examples of summaries generated from Weighted Sentence and BERT2BERT [4] approaches, where the user required number of sentences in weighted sentence method is illustrated in the bracket, and *1st Title* denotes the title of the article ranked in the 1st place among the 5 retrieved articles

It should be noted that, the performance of the retrieval summary system is highly replying on the upstream document retrieval (with a MRR of 0.34) due to the nature of the pipelined method. However, compared with end-to-end methods, our model has better interpretability. Furthermore, the summarization dataset was created using the NLTK method, which may not produce high-quality datasets suitable for model training. It is important to recognize that the performance of the BERT2BERT [4] model is heavily dependent on the quality of the dataset used for summarization. Two potential avenues for future work include collecting large-scale, high-quality summarization datasets, or exploring alternative architectures such as Longformer [2] that support 4096 tokens. By utilizing such architectures, it would no longer be necessary to reduce the article length, and the entire article could be used as input. Another potential area for future improvement is the utilization of more sophisticated and larger models that have been pre-trained on text generation tasks. For example, large seq2seq language models like GPT [8] could be employed for this purpose. However, due to current limitations in computational resources, these models are not currently feasible and will need to be explored in future work.

## 5 Conclusion

In this project, we have investigated two distinct categories of summarization, specifically, extractive summarization utilizing NLTK and abstractive summarization using the BERT2BERT [4] model. The NLTK approach provides a simplistic, swift, and resource-efficient technique for implementing a summarizer without the requirement for training data and extensive computational resources. However, its performance is below optimal in terms of semantic significance and content coverage since it only selects essential sentences from the body text and concatenates them. To mitigate these limitations, we further explored the BERT2BERT [4] approach, aiming to generate summaries that contain more cohesive and coherent information, as it has the potential to create new sentences not present in the original text. By leveraging transfer learning, we utilized the pre-trained knowledge in SciBERT to capture the essence of the article and accurately convey the main idea. However, as previously stated, the primary challenge is the restricted input sequence length of the BERT2BERT [4] model, necessitating the selection of representative sentences using the Weighted Sentence method to shorten the input length for BERT2BERT [4]. Additionally, the constraints imposed by limited computational resources pose a significant obstacle to tuning hyperparameters and limit our capacity to consider models such as Longformer [2] and GPT [8], which are currently impractical to run. Nevertheless, these architectures hold great promise for future experimentation.

## References

1. Beltagy, I., Lo, K., Cohan, A.: SciBERT: A pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3615–3620. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1371, https://aclanthology.org/D19-1371
2. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer (2020)
3. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. Nucleic acids research **32 Database issue**, D267–70 (2004)
4. Chen, C., Yin, Y., Shang, L., Jiang, X., Qin, Y., Wang, F., Wang, Z., Chen, X., Liu, Z., Liu, Q.: bert2bert: Towards reusable pretrained language models (2021)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
6. Jablonski, J.: Natural language processing with python's nltk package. https://realpython.com/nltk-nlp-python/ (2021), (Accessed on 04/10/2023)
7. Joshi, P.: Automatic text summarization using textrank algorithm. https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/? (June 2022), (Accessed on 04/04/2023)
8. Katz, D.M., Bommarito, M.J., Gao, S., Arredondo, P.: Gpt-4 passes the bar exam. Available at SSRN 4389233 (2023)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
10. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Annual Meeting of the Association for Computational Linguistics (2004)
11. van der Meulen, L.: Summarize a text with python — continued. https://towardsdatascience.com/summarize-a-text-with-python-continued-bbbbb5d37adb (November 2022), (Accessed on 04/01/2023)
12. Neumann, M., King, D., Beltagy, I., Ammar, W.: ScispaCy: Fast and robust models for biomedical natural language processing. In: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/w19-5034
13. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
14. Wang, L.L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Burdick, D., Eide, D., Funk, K., Katsis, Y., Kinney, R., Li, Y., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A., Wang, K., Wang, N.X.R., Wilhelm, C., Xie, B., Raymond, D., Weld, D.S., Etzioni, O., Kohlmeier, S.: Cord-19: The covid-19 open research dataset (2020)