GRAPH-BASED EXTRACTIVE SUMMARISATION FOR LONG DOCUMENTS

By

TUBA GOKHAN

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
May 2023

ABSTRACT

The ability to extract the most important information from a longer document or a collection of documents quickly and accurately has always been essential for effective communication and decision-making. By leveraging text summarisation systems, this process can be facilitated more efficiently. Text summarisation help streamline the process by extracting only the essential information from a document or series of documents. Despite significant progress in methods for text summarisation, challenges remain, particularly for unsupervised methods.

This thesis investigates novel unsupervised methods and models in Natural Language Processing (NLP) to improve the quality of text summarisation. Our research makes three major contributions. The first contribution involves improving the performance of sentence similarity detection by combining Deep Learning/Transformer-based models with cluster-based approaches. Our proposed approach improves upon state-of-the-art performance on the Financial News Summarisation (FNS) dataset, indicating its potential for improving the quality of text summarisation. The second contribution explores improving graph models by incorporating more features when calculating node weights. Our proposed approach achieves significant performance gains on four benchmark datasets, demonstrating the potential of incorporating additional features for improving text summarisation. Finally, we propose a novel ranking algorithm for unsupervised graph-based text summarisation. Our proposed algorithm is based on graph centrality measures and can be used to identify the most important nodes in a graph-based summary. We demonstrate the effectiveness of our algorithm through analysis and experiments on four benchmark datasets.

ACKNOWLEDGEMENTS

I am deeply grateful to all those who have supported me throughout my PhD journey. I would like to express my sincerest appreciation to my supervisors, Prof. Mark Lee and Phillip Smith, for their invaluable guidance, insightful feedback, and unwavering support throughout my academic journey. In particular, I would like to extend my special thanks to Prof. Lee for his exceptional support and encouragement behind the academia.

I am also grateful to the Republic of Türkiye Ministry of Education for sponsoring my studies and providing me with the necessary resources to pursue my academic goals. Their support has opened the door for me and enabled me to achieve my dreams.

Furthermore, I would like to express my heartfelt thanks to my beloved mother, Fatma Gökhan, and my precious aunt, Güllü Fidan, for their constant encouragement and unwavering belief in my abilities. They have been a constant source of motivation and inspiration throughout my academic journey.

I am also deeply appreciative of my dear friend, Handan Akgül, for her invaluable insights, unwavering support, and encouragement. Her friendship has been a great source of comfort and inspiration, especially during challenging times.

I want to acknowledge my mischievous cat, Oreo, who has been a constant source of entertainment and distraction during the long hours of research and writing. Despite his misbehaviour, his purring and snuggles have been a source of comfort and stress relief.

Last but not the least, I would like to thank Malcolm Price, for his unwavering support, understanding, and encouragement. His patience, kindness, and unwavering belief in me have been a constant source of inspiration and motivation.

PUBLICATIONS

1. **Tuba Gokhan**, Phillip Smith, and Mark Lee. 2021. Extractive Financial Narrative Summarisation using SentenceBERT Based Clustering. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 94–98, Lancaster, United Kingdom. Association for Computational Linguistics (Gokhan et al., 2021).

2. **Tuba Gokhan**, Phillip Smith, and Mark Lee. 2022. GUSUM: Graph-based Unsupervised Summarization Using Sentence Features Scoring and Sentence-BERT. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 44–53, Gyeongju, Republic of Korea. Association for Computational Linguistics (Gokhan et al., 2022).

3. **Tuba Gokhan**, Phillip Smith, and Mark Lee. 2023. Node-Weighted Centrality Ranking for Unsupervised Long Document Summarization. In: Métais, E., Meziane, F., Sugumaran, V., Manning, W., Reiff-Marganiec, S. (eds) Natural Language Processing and Information Systems. NLDB 2023. Lecture Notes in Computer Science, vol 13913. Springer, Cham. https://doi.org/10.1007/978-3-031-35320-8_21 (Gokhan et al., 2023)

Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

In today's digital and physical environments, the sheer volume of written text data is unprecedented. To enable the efficient use of these data, it is important to provide easily searchable and accessible information. However, finding relevant information within this vast corpus of data is challenging and time-consuming. Fortunately, automation using Natural Language Processing (NLP) approaches, such as Automatic Text Summarisation (ATS), can streamline the process. These approaches attempt to extract only the essential information from a document or series of documents, enabling quick and accurate comprehension of the text's primary message(s). By leveraging the power of ATS, we can efficiently navigate large volumes of text data and extract the most valuable information from it. Despite significant progress in the field of text summarisation, challenges remain. This thesis aims to investigate and develop novel techniques for ATS which address some of these challenges.

The remainder of the chapter describes the motivation for the thesis, specifies the problems to be addressed, formulates the research questions and explains how the proposed solutions contribute to the field.

1.1 Motivation

Maybury (1995) provides a definition of what a summary is as "an effective summary distils the most important information from a source (or sources) to produce an abridged version of the original information for a particular user(s) and task(s)." Based on this definition, the primary objective of ATS systems can be defined as generating a brief and compact version of the central themes in the source document (Radev et al., 2002) while minimising redundancy (Vilca & Cabezudo, 2017). Thus, ATS systems enable readers to obtain the essential information from the original document without the need to read through the entire text (El-Kassas et al., 2021; Nazari & Mahdavi, 2019).

ATS systems have different categories based on various perspectives. (These categories are detailed in Chapter 2.1.) Based on the output summary, summarisation can use either *Extractive Summarisation* or *Abstractive Summarisation* methods (Bae et al., 2019; Erkan & Radev, 2004; Gehrmann et al., 2018; Mallick et al., 2019; Nallapati et al., 2016). Extractive summarisation extracts salient sentences from the original document only, while abstractive summarisation may use different words or phrases to compose the summary. Extractive summarisation is currently more widely used as it is generally more feasible to apply than abstractive summarisation. Another way of categorising ATS is based on the document length into short and long document summarisation (Nallapati et al., 2016; Narayan et al., 2018b). Short documents usually contain generic text, such as news articles, while long documents are usually domain-specific articles, such as scientific papers containing complex formulas and terminologies. On the other hand, ATS can be categorised into supervised and unsupervised methods. Typically, supervised approaches use summaries created by humans to build their models, whereas unsupervised approaches determine relevant parameters without regard to summaries created by humans.

In ATS, supervised methods have proven to be successful. However, the success of these methods is contingent on the availability of large-scale, human-generated summaries which are generally both highly costly and challenging to obtain. In addition, these methods experience difficulties with long document summarisation due to the input length limitation. Intuitively, long document summarisation is harder than short document summarisation due to the significant difference in the number of lexical tokens and breadth of content between short and long documents (Koh et al., 2022). As the length increases, the content that would be considered important will also increase, resulting in a more challenging task for an automatic summarisation model to capture all salient information in a short text summary.

This thesis centres on unsupervised approaches for extractive summarisation of long documents. The research explores the use of graph-based techniques to construct innovative semantic graph models that better represent the document and develop novel ranking algorithms to generate higher-quality summaries. As a result, we propose a novel unsupervised graph ranking approach for extractive summarisation of long documents.

1.2 The Current Limitations of Automatic Text Summarisation

In this section, we outline the main challenges addressed in this thesis with respect to text summarisation.

1.2.1 Long Document Summarisation

Academic articles and business reports are typically classified as "long" due to their significant number of lexical tokens. However, the definition of "long document" in text summarisation is a relative term. In the context of machine learning, a document is considered long when current state-of-the-art models for a normal document cannot be directly implemented in an effective manner due to hardware and model limitations. For instance, Celikyilmaz et al. (2018) considers CNN/DM and NYT benchmark datasets in the news domain as long documents, whereas Koh et al. (2022) defines them as short document datasets. Because the majority of existing state-of-the-art summarisation systems are limited to only 512 to 1024 lexical tokens (Devlin et al., 2019; J. Zhang et al., 2020), a benchmark dataset with an average source document length that exceeds 3,000 lexical tokens is currently regarded as a "long documents" datasets (Manakul & Gales, 2021; Zaheer et al., 2020). Many recently proposed methods incorporate supervised techniques to achieve greater levels of performance. As stated previously, however, these supervised methods cannot be easily applied to long documents (Manakul & Gales, 2021; Meng et al., 2021; Zaheer et al., 2020). These limitations cannot be easily overcome without novel techniques to assist current architectures to reason over a long range of textual inputs. In addition, another important limitation of supervised methods is that they need large-scale, domain-specific and high-quality training datasets together with their gold-standard summaries (Zheng & Lapata, 2019). Unfortunately, datasets with these attributes often do not exist or are costly and time-consuming to create. This makes it difficult to summarise documents outside of the domain or in domains that do not have gold standard summaries.

Unsupervised methods can overcome these difficulties since they do not require domain-specific training datasets. Also, in most cases, they do not have the same length limitations on documents as supervised methods. In order to avoid the issues associated with long scientific document summaries, we focus on unsupervised methods in this thesis.

1.2.2 Graph-Based Summarisation

Graph-based ranking algorithms are a class of algorithms that evaluate the significance of a vertex in a graph by considering global information that is computed recursively from the entire graph instead of relying exclusively on vertex-specific local information (Ramirez-Orta & Milios, 2021). These algorithms are successfully employed in various domains, including social networks and citation analysis. Specifically, graph-based ranking algorithms are a technique for assessing the importance of a vertex in a graph based on information derived from the graph's structure. Similar reasoning underlies the use of graph ranking algorithms in various NLP applications, where information derived from the entire text is used to make local selection and sorting decisions, such as the extraction of key-phrases, word sense disambiguation, and extractive summarisation.

Graph-based extractive summarisation methods generally involve two steps: graph generation and graph analysis (Mihalcea & Tarau, 2004). In the first step, the vertices representing the sentences in the graph are defined, and the edges showing the similarity between these sentences are created. In the second step, the sentences to be included in the summary are determined by calculating the graph centrality score. This score identifies the most important sentences in the graph, and the sentences with the highest score are added to the summary.

The similarity measure plays a crucial role in the graph-based approach and has a direct impact on the ranking results. Researchers continue to work on improving methods for calculating sentence similarity scores in graph-based methods. However, unsupervised graph-based methods based on this approach have been observed to suffer from selecting redundant sentences and generating summaries with unnecessary information (Koh et al., 2022; Ouyang et al., 2009). This problem can be attributed to the fact that most of the existing graph-based algorithms in text summarisation are solely based on similarity and that the importance of sentences in the document are not adequately considered. Therefore, developing better graph models and more effective centrality ranking algorithms that can yield high-quality summaries is essential. In this thesis, we aim to address this problem by proposing novel graph models that consider semantic information and designing more efficient centrality ranking algorithms to generate better summaries.

1.3 Problem Definition and Research Questions

The preceding sections discussed the limitations of supervised approaches on long-document summarisation and the relatively poor performance of graph-based approaches on text summarisation in general. In this thesis, the main objective is to explore novel graph-ranking algorithms that can overcome the problems associated with their application to long document summarisation performance that can better capture the meaning of the documents. In line with this, the research presented in Chapter 3 will focus on the sentence similarity problem. Chapter 4 will look at problems with the graph models that are currently used to map the document while Chapter 5 explores graph ranking algorithms. In this section, we provide a description of the research problems and related research questions.

1.3.1 Sentence Similarity

Sentence similarity in NLP refers to the measure of the degree of semantic similarity or relatedness between two or more sentences (Farouk, 2019; Sarkar et al., 2015). Numerous NLP applications require the use of sentence similarity measurements, including semantic search (Farouk et al., 2019), question answering (De Boni & Manandhar, 2003),and sentiment analysis (Kumar & Jain, 2015). These applications demonstrate that the computation of sentence similarity has become an integral part of knowledge representation.

Graph-based document summarisation methods heavily employ sentence similarity measurements to define relationships. In these methods, graphs are constructed to establish the relationship between sentences. The relationship between any two sentences is represented by an edge, and the weight of each edge is calculated using a similarity measure. There are two ways to measure the similarity between sentences: lexical similarity, which focuses on surface-level similarities in the words used, and semantic similarity, which considers the meaning conveyed by the sentences, even if they use different words. However, due to the large variety of natural language expressions, identifying semantically similar statements can be challenging (Achananuparp et al., 2008). Especially in ATS, failure to accomplish this causes redundancy. In the context of ATS, redundancy refers to the repetition of information or ideas within a summary. This occurs when multiple sentences or phrases in the summary convey the same or similar information, resulting in unnecessary duplication of content. Redundancy reduces the overall effectiveness of a summary, as it can make the summary longer and more challenging to read while also potentially missing out important information that could have been included instead.

Redundancy removal is a necessary sub-task of ATS because it helps to eliminate duplicated or similar information. Identifying similar sentences inside the documents is the most significant part of a redundancy removal strategy (Sarkar et al., 2015). For this reason, we hypothesise that the performance of graph-based summarisation can be improved by enhancing the similarity measure thus helping to reduce redundancy. This leads us to the first research question examined in this thesis:

Research Question 1: How do specific sentence similarity measures affect the redundancy removal process in text summarisation?

To address this question, in Chapter 3 we focus on investigating pre-trained language models to provide a better measure of the similarity of sentences. Next, we research how to use pre-trained language models to detect semantically similar sentences. Finally, we develop a cluster-based summarisation system to evaluate the effect of these models on text summarisation.

1.3.2 Graph-Models for Extractive Summarisation

A graph is a collection of points, called vertices or nodes, that are connected by lines, called edges. Graph theory provides a framework for analysing the structure and properties of graphs and developing algorithms and models based on graph representations. Graph-Based summarisation approaches are based on the idea of graph theory. In these approaches, a graph's points (vertices or nodes) represent sentences (or words), and the edges show how sentences relate to each other. Centrality is a measure used in graph theory to determine the importance or significance of a node, or vertex, within a graph (Faudree, 2003). The centrality scores of nodes can be used to construct summaries using ranking algorithms (Bichi et al., 2022).

In graph theory, node weights play an important role and can be used to drive the behaviour of graph-based algorithms and analyses. Node weights are values assigned to nodes that represent some characteristic of the nodes, such as their importance, relevance, or significance. The definition of node weights in graphs depends on the specific application and the algorithm being used. For example, in text summarisation, node weights might represent the relevance of each sentence to the overall topic or theme, which can be used to generate a summary that emphasises the most important information. However, in the majority of the previous research in text summarisation using edge-weighted graphs, weights for the nodes were assumed to be uniform. The difficulty and complexity of identifying the mapping between node characteristics is the primary reason for disregarding node weights (Singh et al., 2020). As a result, there is a potential missed opportunity to use informative node weights to improve the performance of graph based methods for text summarisation.

This leads to the second research question that we investigate in this work:

Research Question 2: What advancements in graph models significantly improve the representation of semantic information for text summarisation?

To address this question, in Chapter 4 we focus on investigating node-weighted graphs and sentence features in order to represent more knowledge. Next, we research defining node weight via sentence features scoring to extract the salient sentences for summaries. Finally, we develop a graph-based summarisation system to evaluate the effect of these models on text summarisation.

1.3.3 Graph Ranking Algorithms for Extractive Summarisation

As stated in Section 1.3.2, calculating node centrality is a critical aspect of graph analysis. Centrality is evaluated using degree or ranking algorithms. In extractive text summarisation, the most important sentences typically have the highest centrality scores, and these sentences are extracted as a summary. However, traditional unsupervised graph-based methods have a tendency to select repetitive sentences, resulting in summaries that lack variety (Koh et al., 2022). There are several reasons for this problem. First, the methods used to measure similarity are not comprehensive enough for redundancy removal (Section 1.3.1). Secondly, node weights are inadequately defined (Section 1.3.2). Another reason is the centrality ranking algorithms employed in models assume that a sentence is more significant if it is similar to other sentences (Liang et al., 2021). This idea generally works well for single-aspect documents. However, there are always multiple aspects, especially in long documents.

To overcome the above-mentioned problems, a fully weighted graph representation of the document should be created, and effective ranking algorithms should be designed for the analysis of this graph. This kind of approach can enable a more comprehensive and precise representation of semantic information, enhancing text summarisation tasks' performance. This leads to the third research question that will be answered by this thesis:

Research Question 3: How does the implementation of node-weighted graph ranking algorithms specifically impact the summarisation of long documents?

To address this question in Chapter 5, we focus on investigating graph ranking algorithms and sentence centrality methods in order to overcome the limitations of current text summarisation systems. Next, we research combining current sentence centrality methods with node-weighted graph models. We develop a node-weighted graph ranking algorithm for unsupervised extractive long document summarisation. Finally, we present a system to evaluate the effect of this algorithm on document summarisation.

1.4 Contributions

This thesis offers the following contributions based on the research questions:

- We first concentrate on solving the sentence similarity problem in long documents using the Bidirectional Encoder Representations from Transformers (BERT) architecture. We offer an ATS based on cluster-based approaches to measure of effectiveness of SentenceBERT models on similarity problem. To evaluate our system's performance, we select a long document dataset of UK annual reports for corporations listed on the London Stock Exchange (LSE) from 2002 to 2017. These reports typically consist of approximately 80 pages on average, with some exceeding 250 pages. The results show that SentenceBERT models help to mitigate the sentence similarity problem in text summarisation (see Chapter 3). These results were presented at FNS 2021 (Gokhan et al., 2021).

- We then describe an improved approach to more accurate analysis of graphs for text summarisation. We focus on node-weighted graph models using feature sentence scoring methods. We first define the sentence feature scores represented at the vertices, indicating the importance of each sentence in the document. After this stage, we use Sentence-BERT models to obtain sentence embeddings to capture the meaning better. In this way, we define the edges of a graph where semantic similarities are represented. Next, we create an undirected graph that includes sentence significance and similarities between sentences. In the last stage, we determine the most important sentences in the document with the ranking method on our graph model. Experiments on CNN/Daily Mail, New York Times, arXiv, and PubMed datasets show that our approach achieves high performance for unsupervised graph-based summarisation when evaluated against other automatically generated and also human generated summaries (see Chapter 4). These results were presented at TextGraph-16 held during COLING 2022 (Gokhan et al., 2022).

- Finally, we examine the problem of extracting the most salient sentences in long documents by proposing a node-weighted graph ranking algorithm. We propose a novel centrality ranking algorithm. In our approach, we create an undirected, fully weighted graph model for each document. First, to define augmented node-weight (i.e., sentence), we use two methods: Latent Semantic Analysis and Sentence Feature Scoring. Secondly, we define the edge-weighted (i.e., similarity) and employ Sentence-BERT to compute sentence similarity. Thirdly, we apply our ranking method, developed based on eigenvector centrality, by including node weights. Finally we experimentally evaluate our proposed algorithm. The experimental results of our system prove that our straightforward unsupervised method shows performance equivalent to that of state-of-the-art supervised neural models trained on hundreds of thousands of samples of large documents (see Chapter 5). These results were presented at NLDB 2023 (Gokhan et al., 2023).

1.5 Thesis Organisation

The remainder of the thesis is organised as follows: In Chapter 2, several research areas of ATS are presented. This is followed by exploring terminologies related to summarisation, including main tasks and sub-tasks of summarisation systems. In addition, related works on unsupervised, cluster-based, and graph-based summarisation systems are reviewed. Finally, the details of various NLP techniques, including classical methods and pre-trained models, are summarised as they are utilised in our research.

In Chapter 3, the details of the measurement of sentence similarity using Sentence-BERT models for ATS are presented. This chapter also provides the experimental setup, baselines, and empirical results using these models on annual financial reports. The chapter concludes by presenting the empirical results of our extractive unsupervised cluster-based summarisation system.

Chapter 4 describes a novel node-weighted graph model for text summarisation. The chapter also presents the experimental setup, baselines, and empirical results using sentence feature scoring methods on that graph model. The chapter concludes with the empirical results of our graph model on four benchmark datasets.

In Chapter 5, a novel graph-ranking algorithm for long-document summarisation is presented. The chapter also explains the model settings, baselines, experiments, and evaluation results on two long-document datasets. Finally, the chapter wraps up with a discussion of the effect of the node-weighted graph model and centrality ranking algorithm for long-document summarisation.

Lastly, Chapter 6 provides concluding remarks and ends the thesis. This final chapter highlights the contributions of the current study and potential areas for future research.

Chapter 2

Related Work

In the previous chapter, we described the motivation for the thesis, specified the problems to be addressed, formulated the research questions and explained how the proposed solutions contribute to the field. In this chapter, we present the recent literature in the research area of text summarisation. Section 2.1 gives an overview of text summarisation, while Section 2.2 provides information the different tasks required for text summarisation. The final Section (2.3) presents a discussion of text summarisation methods.

2.1 Text Summarisation: An Overview

The research on ATS started in 1958 when Luhn developed a algorithm for summarising technical and journal articles (Luhn, 1958). Even though there have since been many advancements in the field, researchers still work to develop an ATS system that can meet three essential criteria (Gambhir & Gupta, 2017). First, the system should cover all the significant topics included in the input text. Secondly, it should eliminate redundant information. Thirdly, the summary should be coherent, consistent and readable.

ATS presents various tasks for researchers, including the need to: 1) identify the most informative segments in the input text for inclusion in the generated summary, 2) summarise long documents, such as books, 3) summarise multiple documents, 4) evaluate the computer-generated summaries without requiring comparison to a human-generated summaries, and 5) generate an abstractive summary that closely resembles a human-generated summary (El-Kassas et al., 2021). To address these tasks, the analysis of ATS is organised into categories as illustrated in Figure 2.1. The information in Figure 2.1 is compiled from the studies prepared by Gambhir and Gupta (2017), Rahimi et al. (2017)

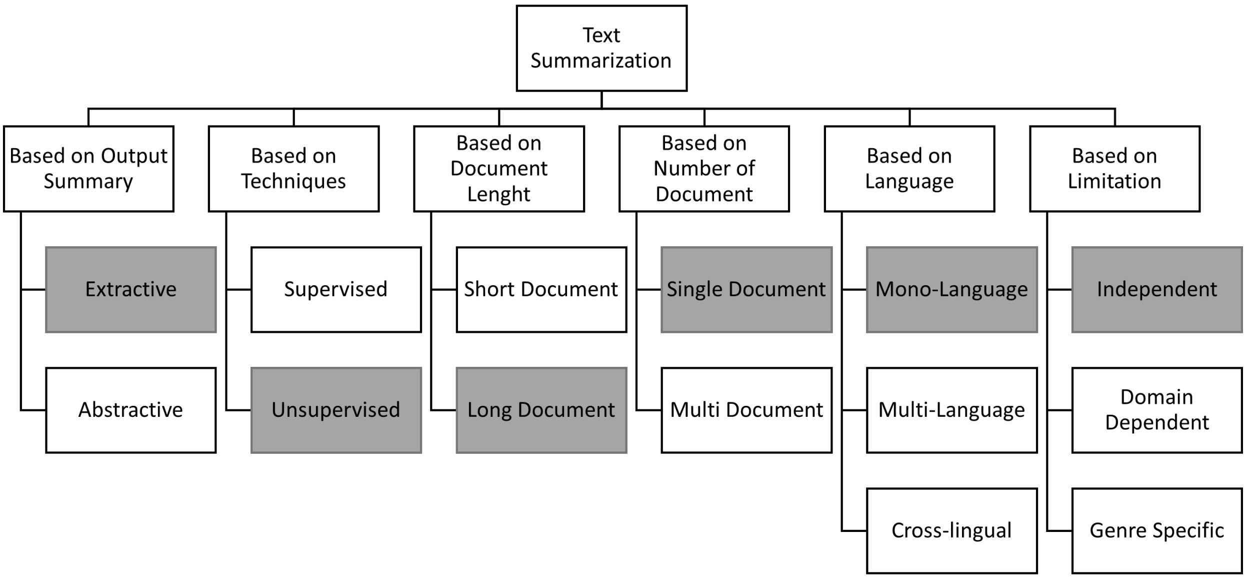


Figure 2.1: Categories of automatic text summarisation systems.

As previously discussed in Section 1.1, our research centres on Extractive, Unsupervised, and Long-Document summarisation. Additionally, we base our work on Single-Document and Mono-Language approaches. Regarding the category of based on limitations, our work falls under the category of independent summarisation as it aims to produce general summaries that are not domain or genre-specific.

2.2 Document Summarisation Tasks

The domain of ATS encompasses various applications such as opinion summarisation (Basu Roy Chowdhury et al., 2022), transcript summarisation (Song et al., 2022), dialogue summarisation (Y. Zhang et al., 2022), code summarisation (Nie et al., 2022), question-answer summarisation (Deutsch & Roth, 2022) , and document summarisation. This thesis centres on document summarisation, with a specific focus on three tasks: Extractive summarisation, summarisation of long documents and measuring sentence similarity. This section provides detailed information about these tasks.

2.2.1 Extractive Text Summarisation

Extractive approaches are faster to compute and less complicated than the abstractive approaches. While abstractive approaches can be more brief and adaptable, extractive approaches better preserve the original's tone and are typically more fluent (Dong et al., 2021; Kryscinski et al., 2020). They produce more accurate results due to their direct extraction of sentences, which ensures that the summary includes the precise terminologies found in the original text (Tandel et al., 2016). As a result, readers can read the summary with a clear understanding of the content. Figure 2.2 displays the general architecture of the extractive text summarisation system, which involves three main stages (El-Kassas et al., 2021).

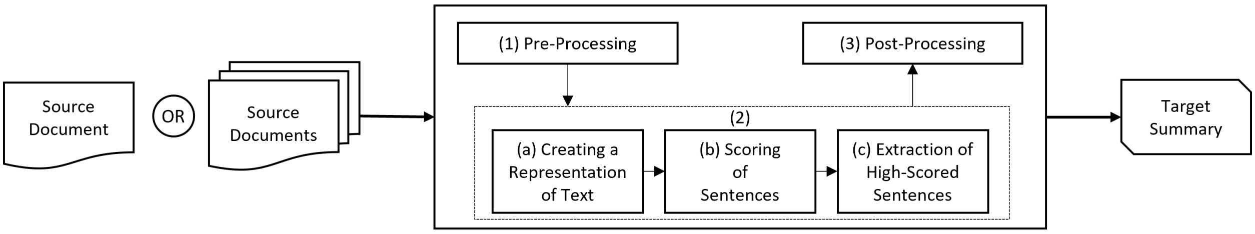


Figure 2.2: The general architecture of extractive text summarisation systems.

The first stage is the pre-processing of the input text. The second stage includes the following processing tasks: (a) creating a representation of the input text using methods like N-gram, bag-of-words, or graphs, (b) scoring the sentences based on this representation, and (c) selecting the most significant sentences from the input document(s) and combining them to form a summary. The summary's length depends on a preferred compression rate. The third stage, post-processing, involves rearranging the selected sentences, substituting pronouns with their antecedents, and replacing relative temporal expressions with actual dates.

In the literature, there exists a range of extractive text summarisation methods. During our research, we focus on four methods: Clustering-Based, Statistical-Based, Semantic-Based and Graph-Based.

Clustering-Based methods use an extractive summariser to determine the significance of a sentence based on its proximity to the cluster centroid. This technique considers both the relevance of the sentences, and redundancy elimination, when generating a summary (Nazari & Mahdavi, 2019). The sentence selection process for this method involves the following steps: 1) generating vector representations (feature vectors) of sentences as input, 2) clustering the input vectors using an appropriate algorithm, 3) ranking and arranging the clusters in a manner that prioritises those with more important sentences, and 4) selecting representative sentences for the summary from these clusters.

Statistical-Based Methods involve the extraction of important sentences and words from the source text through statistical analysis of a range of features. Identification of the "most significant" sentences uses features such as sentences' position, length, frequency of thematic words, and frequency of numerical data (El-Kassas et al., 2021). The sentence scoring process for a statistical-based extractive summariser encompasses two steps (Gambhir & Gupta, 2017): 1) the selection and computation of various statistical and/or linguistic features and the assignment of weights to them, and 2) the assignment of a final score to each sentence in the document via a feature-weight equation.

Semantic-Based Methods refer to a set of extractive ATS techniques that use the meaning and context of the text to identify important sentences. Latent Semantic Analysis (Nenkova et al., 2011) is a popular unsupervised method that represents the semantics of text based on word co-occurrence. The process of scoring sentences in an LSA-based extractive summariser involves creating an input matrix and applying Singular Value Decomposition to identify relationships between terms and sentences. Other semantic-based techniques, such as Semantic Role Labelling (SRL) (Gildea & Jurafsky, 2002) and Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007), are also used for ATS.

Graph-based methods in text summarisation involve creating a graph where each node represents a sentence in the input text, and the edges between nodes indicate the similarity between sentences (El-Kassas et al., 2021). Similarity can be defined based on different techniques such as lexical similarity, semantic similarity, or context overlap. The graph-based approach involves two main steps: The first step is to build the graph. The sentences are preprocessed and represented as nodes in the graph. Then, the edges between the nodes are created based on the similarity between sentences. In the second step, the most important sentences are identified by ranking the nodes in the graph. The final summary is generated by selecting the top-ranked sentences in the graph, which represent the most important information in the input text.

Table 2.1, based on the works of Gambhir and Gupta (2017) and El-Kassas et al. (2021), provides an overview of the advantages and disadvantages of the most commonly used methods. Based on these, we initially develop our research within the framework of clustering-based methods. Subsequently, we advance our methodology through a hybridisation of statistical & semantic methods and graph-based methods to attempt to mitigate the drawbacks of graph-based approaches.

Table 2.1: Pros and cons of the extractive text summarisation methods.

Method	Pros	Cons
Clustering Based	Sentence repetition can be minimised with the use of these methods, which group similar sentences together, making it a suitable technique for multi-document summarisation since it can group sentences related to the same topic from different documents.	Prior specification of the number of clusters is necessary for these methods, and even though highly scored sentences may be similar, redundancy removal techniques are required; in addition, while some sentences may express more than one topic, each sentence can only be assigned to a single cluster..
Statistical Based	They demand low processing resources and memory, and do not necessitate additional linguistic knowledge or sophisticated linguistic analysis. They are independent of language.	Important sentences could potentially be excluded from the summary if they receive a lower score compared to other sentences. Conversely, similar sentences with high scores may be included in the summary.
Semantic Based	are able to comprehend the underlying meaning and context of the input text, resulting in higher quality and more relevant summaries, and they are also independent of language.	The quality and relevance of the generated summary are closely linked to the quality of the semantic representation of the input text, with techniques like LSA often consuming a significant amount of time to compute SVD.
Graph Based	can enhance coherence in the summary by considering the similarity between sentences. Can also detect and remove redundant information. Additionally, they are language and domain-independent, making them applicable to various types of input texts.	These methods generally assume that all sentences (words) have equal weights, disregarding their importance in the document. Additionally, these methods may fail to recognise semantically equivalent sentences due to the similarity measurements. The similarity computation's accuracy influences the sentence selection in the final summary.
Machine Learning Based	A large set of training data is needed to enhance the selection of sentences for summarisation, while relatively simple regression models can yield better outcomes compared to other classifiers.	A large dataset of manually-created extractive summaries is necessary for the labelling of each sentence in the original training documents as either "summary" or "non-summary".
Deep Learning Based	The training of the network can be customised to match the preferences of the human reader, while the feature set can be modified according to the specific needs of the user.	The creation of training data for neural networks requires considerable manual effort, and both the training and testing phases can be time-consuming. Additionally, it can be difficult to understand how the network makes its decisions.

2.2.2 Long-Document Summarisation

Summarisation of long documents presents a more complex task than for short documents, owing to the vast differences in the number of lexical tokens and content breadth. The increased length of documents further compounds the difficulty of capturing all crucial information within the summary's constrained output length. Additionally, long documents more often contain specialised scientific content with complex formulas and terminologies, whereas short documents are commonly generic, like news articles. To clarify the distinction between short and long documents, Koh et al. (2022) proposed a conceptualisation of the summarisation task problem based on three fundamental aspects: length of document, breadth of content, and degree of coherence.

- **Length of Document:** In machine learning and deep learning, a document is considered long when it exceeds the capacity of current state-of-the-art models due to hardware and model limitations. A benchmark dataset with an average source document length of over 3,000 lexical tokens can currently be considered to be a "long document" because existing summarisation systems are limited to 512-1024 tokens. Novel techniques are required to assist current architectures in processing long inputs.
- **Breadth of Content:** The length of a document is usually proportional to the amount of information it contains. Koh et al. (2022) demonstrates that the relative length of the summary decreases exponentially as the length of the source document expands. However, the length of a summary must be limited to what the user considers acceptable. Due to this limiting factor, for long documents the length of the summary must be much shorter compared to the length of the source document, making it difficult to include all of the important information. Additionally, user preferences and expectations become more diverse, making it even more challenging to create a satisfactory summary (Kryscinski et al., 2019).

- **Degree of Coherence:** Long documents are often structured into sections to help readers understand them. However, each section may have a different topic, even though they are all related to the central narrative of the document. This makes summarising long documents more challenging because summary models need to ensure that the summary is fluent, non-redundant, and semantically coherent, even when concatenating important information from different sections.

Researchers have created benchmark datasets to assess the effectiveness of summarisation approaches. These datasets have different characteristics that affect the performance of the models, the suitability of the summarisation approach (extractive or abstractive), and the effectiveness of evaluation metrics. In our research, we examine datasets based on document length. Table 2.2 shows the most commonly used datasets and their statistics.

Table 2.2: Comparison of short and long document summarisation datasets.

	Short Document Datasets						**Long Document Datasets**			
	CNN/DM	NYT	NWS	XSum	WikiHow	Reddit	ArXiv	PubMed	BigPatent	BillSum
# **document**	312k	1.8m	1.3m	227k	230k	123k	216k	133k	1.3m	22k
# **summary tokens**	55	96	31	24	70	23	242	208	117	243
# **document tokens**	774	1109	767	438	501	444	6446	3143	3573	1686
# **summary sentences**	3.8	1.2	1.5	1	5.3	1.4	6.3	7.1	3.6	7.1
# **document sentences**	29	32	31	19	27	22	251	102	143	42

Short Document Datasets. Table 2.2 shows datasets of short documents, including four popular news datasets (CNN-DM (See et al., 2017), NYT (Sandhaus, 2008), NWS (Grusky et al., 2018), and XSum (Narayan et al., 2018b)) and datasets from other domains (Reddit-TIFU (Kim et al., 2019) and WikiHow (Koupaee & Wang, 2018)). The news datasets consist of news articles and their corresponding human-created summaries or summaries created by combining highlighted sentences. The Reddit-TIFU dataset is collected from the subreddit r/TIFU, and uses the title of topic as the summary, while the WikiHow dataset uses the first sentence of each paragraph as the summary.

Long Document Datasets. Table 2.2 also presents the widely used datasets for long document summarisation research, including arXiv, PubMed, BIGPATENT, and BillSum. arXiv and PubMed are datasets that contain scientific papers collected from arXiv.org and PubMed.com respectively(Cohan et al., 2018). They are among the earliest datasets used for large-scale long document summarisation research and they use abstracts of articles as summaries. BIGPATENT (Sharma et al., 2019) is a huge dataset with over 1.3 million U.S. patent documents along with human-written summaries using the abstractive approach. BillSum (Kornilova & Eidelman, 2019) is a dataset that focuses on summarising Congressional and California state bills, which have different content structures and writing styles from other domains. Summaries of all these long document datasets are created as abstractive and human generated.

Table 2.2 shows that long documents have a higher compression ratio in their summaries than short documents. The main reason for this may be that there is more redundant information in the source document. However, this higher compression ratio may also have caused more information to be lost in the summarisation process. Therefore, it is more difficult to summarise long documents as the model must find the key narrative while excluding less important content.

In our research, we develop an approach that considers the challenges involved in summarising long documents. To assess the effectiveness of our approach, we use two long document datasets (arXiv and PubMed) and two short document datasets (CNN/DM and NYT). We select these datasets because they have been available for a long time, making it easier to compare the performance of our approach over time.

2.2.3 Measuring Sentence Similarity

The measurement of sentence similarity is the subject of ongoing research, leading to the development of various techniques. Vector space models (VSMs) have become a popular method for computing sentence similarity in recent years. These models were first introduced by Salton et al. (1975) and work by analysing the relationship between vectors that represent data, enabling the comparison of the similarity of two vectors from a geometric perspective. Typically, the measurement of similarity between two vectors is represented as the distance between them on a scale ranging from 0 to 1. This distance is commonly known as the similarity score. Word embedding and sentence embedding approaches belong to the family of vector space models (VSMs) and are used to represent words as vectors in a high- dimensional space. The vectors are created by analysing the co-occurrence patterns of words in a large collection of texts. The embedding techniques investigated in our research are presented in this section.

Word Embeddings

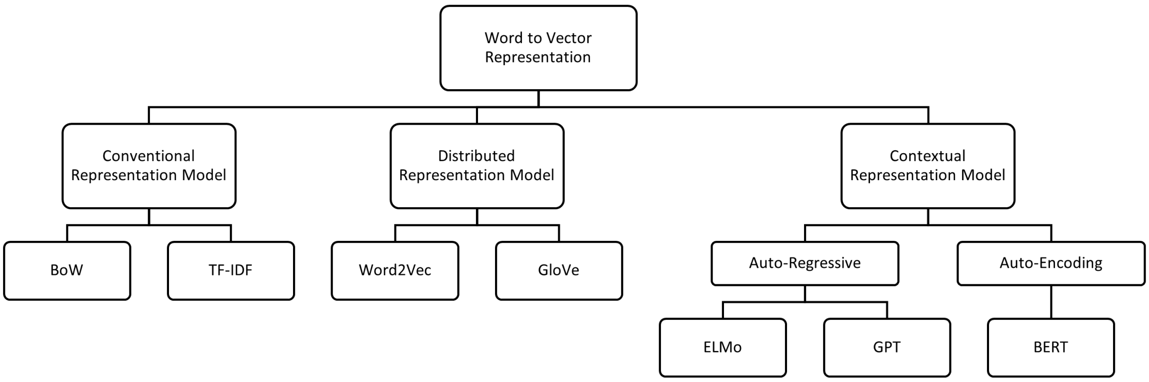

Word to Vector Representation
Conventional Representation Model
Distributed Representation Model
Contextual Representation Model
BoW
TF-IDF
Word2Vec
GloVe
Auto-Regressive
Auto-Encoding
ELMo
GPT
BERT

Figure 2.3: The main word embedding techniques used to create a word to vector representation.

Figure 2.3 illustrates the three main categories of word embedding techniques: conventional, distributional, and contextual (Asudani et al., 2023). The conventional models, also known as count-based or frequency-based models, are further subdivided into bag of words (BoW) and term frequency-inverse document frequency (TF-IDF) models. The BoW model creates a frequency distribution of each word in the document, which is used to create a vector representation of the document. This vector represents the document in a high-dimensional space, where each dimension corresponds to a word in the vocabulary. The value in each dimension represents the frequency of the corresponding word in the document. In TF-IDF model, the TF (term frequency) part of the equation measures how frequently a word appears in a document, while the IDF (inverse document frequency) part measures how rare or common the word is in a collection of documents. The product of these two values gives a weight to each word, which can then be used to create a vector representation of the document.

Distributional models, which are also referred to as static word embedding models, are based on probabilistic-distributional models, such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) model. Word2Vec is a neural network-based algorithm that learns to represent words as vectors in a high-dimensional space based on their co-occurrence patterns in a corpus of text. The Word2Vec algorithm uses two main architectures: the Continuous Bag-of-Words (CBOW) and the Skip-Gram model for training. In the CBOW architecture, the algorithm predicts a target word based on its context (surrounding words), while in the Skip-Gram model, the algorithm predicts the context words given a target word. After the Word2Vec model is trained, each word is represented as a vector. GloVe (Global Vectors for Word Representation) is another technique that is trained by analysing global co-occurrence statistics of words in a corpus. The algorithm combines the count-based model (e.g., co-occurrence matrix) and the prediction-based model (e.g., Word2Vec) by constructing a co-occurrence matrix from the text and then using matrix factorisation techniques to learn word embeddings.

Lastly, the contextual word embedding models are grouped into two subcategories: auto-regressive and auto-encoding models. These include Embeddings from Language Models (ELMo), Generative Pre-trained Transformer (GPT), and bidirectional encoder representations from transformers (BERT) models. ELMo (Peters et al., 2018) generates word representations based on their context in a given sentence. It uses a bi-directional language model that takes into account the entire sentence to create word embeddings that are sensitive to the context in which the word is used. ELMo, also, uses a character-based CNN architecture to generate word embeddings that capture the morphology of the word as well as its context. GPT (Radford et al., 2018) also generates word representations based on their context. It uses a transformer-based language model that generates word embeddings and an unsupervised learning approach to pre-train a deep neural network. Unlike ELMo, GPT uses a uni-directional language model that generates word representations based only on the previous words in the sentence. BERT (Devlin et al., 2019) uses Masked Language Modelling (MLM) to pre-train the model. During training, some words in the input text are randomly masked and the model has to predict the masked words based on their context. This process allows the model to learn contextualised representations of words that take into account their meaning in a particular sentence or passage. One of the important features of BERT is its ability to handle bidirectional context. BERT evaluates the entire sequence of words in a sentence or passage when making predictions.

Sentence Embeddings

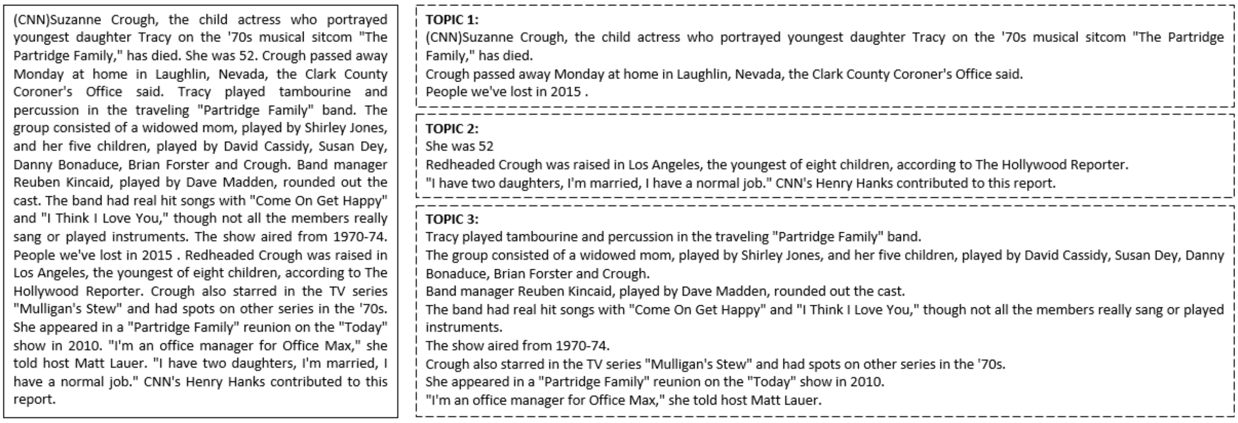
While word embeddings represent individual words as vectors in a high-dimensional space, sentence embeddings represent entire sentences as vectors in a high-dimensional space. Sentence embeddings are created by taking the word embeddings of each word in the sentence and combining them. Thus, the sentence vector captures the overall meaning and context of the sentence, rather than just the individual words. The main difference between word embeddings and sentence embeddings is the level of detail. Word embeddings focus on individual words and their relationships, while sentence embeddings focus on the overall meaning and context of a sentence. SentenceBERT (Reimers & Gurevych, 2019) and RoBERTa (Zhuang et al., 2021) are powerful models for generating sentence embeddings. SentenceBERT, or SBERT, is a variant of the BERT architecture. It uses a siamese network architecture to compare the similarity between two sentences and produces a fixed-length vector representation for each sentence. RoBERTa is another variant of the BERT model that is pre-trained on a larger and more diverse corpus of text than BERT. RoBERTa uses a similar architecture to BERT but makes a few modifications to the pre-training process.

2.3 Related Methods

The research detailed in this thesis centres on unsupervised and extractive techniques for document summarisation. As explained in Section 2.2.1, our focus is on clustering and graph-based methods. This section briefly overviews clustering and graph-based unsupervised and extractive methodologies.

2.3.1 Clustering Based Methods

The process of defining topics and scoring sentences based on these topics is an essential aspect of document summarisation. Sentence clustering can be used to try to identify the topics present in a document, thus making it a popular approach in document summarisation (Pawar et al., 2022). Topic refers to a subject or theme that is discussed in the sentences or documents being analysed. Figure 2.4 shows a sample document from CNN/DM dataset and illustrative example of how the topics within it could be envisioned.



(CNN)Suzanne Crough, the child actress who portrayed youngest daughter Tracy on the '70s musical sitcom "The Partridge Family," has died. She was 52. Crough passed away Monday at home in Laughlin, Nevada, the Clark County Coroner's Office said. Tracy played tambourine and percussion in the traveling "Partridge Family" band. The group consisted of a widowed mom, played by Shirley Jones, and her five children, played by David Cassidy, Susan Dey, Danny Bonaduce, Brian Forster and Crough. Band manager Reuben Kincaid, played by Dave Madden, rounded out the cast. The band had real hit songs with "Come On Get Happy" and "I Think I Love You," though not all the members really sang or played instruments. The show aired from 1970-74. People we've lost in 2015 . Redheaded Crough was raised in Los Angeles, the youngest of eight children, according to The Hollywood Reporter. Crough also starred in the TV series "Mulligan's Stew" and had spots on other series in the '70s. She appeared in a "Partridge Family" reunion on the "Today" show in 2010. "I'm an office manager for Office Max," she told host Matt Lauer. "I have two daughters, I'm married, I have a normal job." CNN's Henry Hanks contributed to this report.

TOPIC 1:
(CNN)Suzanne Crough, the child actress who portrayed youngest daughter Tracy on the '70s musical sitcom "The Partridge Family," has died.
Crough passed away Monday at home in Laughlin, Nevada, the Clark County Coroner's Office said.
People we've lost in 2015 .

TOPIC 2:
She was 52
Redheaded Crough was raised in Los Angeles, the youngest of eight children, according to The Hollywood Reporter.
"I have two daughters, I'm married, I have a normal job." CNN's Henry Hanks contributed to this report.

TOPIC 3:
Tracy played tambourine and percussion in the traveling "Partridge Family" band.
The group consisted of a widowed mom, played by Shirley Jones, and her five children, played by David Cassidy, Susan Dey, Danny Bonaduce, Brian Forster and Crough.
Band manager Reuben Kincaid, played by Dave Madden, rounded out the cast.
The band had real hit songs with "Come On Get Happy" and "I Think I Love You," though not all the members really sang or played instruments.
The show aired from 1970-74.
Crough also starred in the TV series "Mulligan's Stew" and had spots on other series in the '70s.
She appeared in a "Partridge Family" reunion on the "Today" show in 2010.
"I'm an office manager for Office Max," she told host Matt Lauer.

Figure 2.4: A sample document from the CNN/DM dataset, along with an example of the possible topics that are encompassed within the document.

Clustering-based methods represent a class of unsupervised summarisation techniques used for grouping topics, wherein similar sentences or documents are grouped into clusters, and representative sentences are then selected from each cluster to create the summary. Each cluster is regarded as a distinct topic within the document. While the components of a cluster share a high degree of similarity with each other, they are less similar to components belonging to other clusters. The underlying idea behind this approach is that redundant or overlapping information is typically present in similar sentences or documents, and selecting one representative sentence from each cluster helps to ensure that the most important information is included in the summary while reducing redundancy.

The clustering process starts by representing the sentences as feature vectors. A feature vector is a mathematical representation of the sentence, where each dimension represents a feature or characteristic of the sentence. The features can be simple, such as word frequencies or TF-IDF scores, or more complex, such as semantic embeddings or syntactic structures. Different techniques can be applied to create feature vectors. Table 2.3 presents some clustering-based techniques that are examined for our work.

Table 2.3: Unsupervised extractive cluster-based document summarisation methods.

Reference	**Technique**	**Category**
Qazvinian and Radev (2008)	Citation	Single-Document
Agarwal et al. (2011)	Frequent Term Based	Multi-Document
Chen and Zhuge (2014)	Citation	Multi-Document
Wang et al. (2016)	Word2Vec	Single-Document
Alguliyev et al. (2019)	TF-IDF	Multi-Document
Miller (2019)	BERT	Single-Document
Khan et al. (2019)	TF-IDF	Single-Document
Haider et al. (2020)	Word2Vec	Single-Document
Pawar et al. (2022)	TF-IDF	Multi-Document

Table 2.3 illustrates that the use of cluster-based methods is common in multi-document summarisation due to their capability to group distinct topics into individual clusters. We hypothesise that clustering-based methods effectively summarise long documents because long documents frequently involve multiple topics. In our study, we evaluate the performance of clustering-based methods for summarising long documents. Specifically, we explore the influence of incorporating SentenceBERT models, an advanced technique for generating feature vectors, on the performance of these methods.

2.3.2 Graph Based Methods

Graph-based methods are widely used in unsupervised extractive summarisation because they offer a flexible and efficient way of representing text as a graph. This representation allows the algorithm to capture the overall structure and coherence of the text, which is essential for summarisation. In our work, we investigate several graph-based methods, which are presented in Table 2.4.

Table 2.4: Unsupervised extractive graph-based document summarisation methods.

Reference	**Edge Weight (similarity)**	**Node Weight**	**Graph Type**	**Category**
TextRank (Mihalcea & Tarau, 2004)	Word overlap /co-occurrence	NA	Undirected	Short
LexRank (Erkan & Radev, 2004)	BoW	NA	Undirected	Short
PACSUM (Zheng & Lapata, 2019)	BERT	NA	Directed	Short
STAS (S. Xu et al., 2020)	Transformers	NA	Directed	Short
J. Liu et al. (2021)	BERT	NA	Undirected	Short
FAR (Liang et al., 2021)	BERT	NA	Directed	Short & Long
HIPORANK (Dong et al., 2021)	BERT, SBERT, SRoBERTa	NA	Directed	Long

TextRank (Mihalcea & Tarau, 2004) and LexRank (Erkan & Radev, 2004) are among the most important examples of early studies in Graph-Based extractive summarisation. The main difference between these algorithms is the strategy used to compute sentence similarity (Table 2.4) and generate graph edges. TextRank represents sentences as nodes in an undirected graph and computes edge weights based on sentence occurrence similarity, while LexRank includes a pruning threshold for edges at graph initialisation to remove edges with low weights.

More recently, researchers have continued to develop graph-based methods. PACSUM (Zheng & Lapata, 2019) created a directed graph using BERT (Devlin et al., 2019) to calculate sentence similarities. In the directed graph that PACSUM created, the edges represent the relative position of the sentences in the document. In STAS, S. Xu et al. (2020) pre-train a hierarchical transformer model using unlabelled documents. They design a method to rank sentences using sentence-level self-attentions and pre-training objectives. J. Liu et al. (2021) publish a graph-based method based on both the similarities and relative distances in the neighbourhood of each sentence. They also generalise their approach from single-document summarisation to a multi-document setting by aggregating document-level graphs via proximity-based cross-document edges. FAR (Liang et al., 2021) propose a facet-aware centrality-based model. They introduce a modified graph-based ranking method to filter irrelevant sentences using sentence-document similarity. Dong et al. (2021) propose a long document summarisation approach, called HIPORANK. This approach augments the measure of sentence centrality by inserting directionality and hierarchy into the graph with boundary positional functions and hierarchical topic information grounded in discourse structure.

Table 2.4 indicates that recent research efforts have primarily emphasised BERT-based similarity measurement techniques. Furthermore, none of these methods directly calculate node weights; instead they are implicitly defined based on the edge weights. This thesis explores methods for the direct generation of node weights and proposes an integrated evaluation approach that considers both node and edge weights in the graph.

2.4 Evaluation Metrics

Evaluation of summarisation systems plays a crucial role in the development of summarisation tools because only a well-executed evaluation can measure whether the end-user's needs are being met and can determine whether the system is outperforming existing tools (Iskender et al., 2021). It can also provide valuable feedback to researchers, helping them identify areas for improvement and improve the system over time. However, in text summarisation, the evaluation process is challenging due to the complexity of the task. Furthermore, there may be multiple equally effective summaries for the same source document since not all essential information can fit within a given summary length (Hardy et al., 2019).

Human evaluation is generally regarded as the most reliable method for evaluating the quality of summarisation (Celikyilmaz et al., 2020). However, due to human evaluations' being expensive and time-intensive in nature, researchers frequently use automated metrics, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), JS-2 (Louis & Nenkova, 2013), S3 (Peyrard et al., 2017) or Moverscore (W. Zhao et al., 2019). Out of all the methods used to measure the quality of summarisation, ROUGE is the most widely used.

2.4.1 ROUGE Metric

ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a metric used to measure the quality of summarisation. It counts the number of overlapping word sequences (called n-grams) between a generated summary and a set of reference summaries. There are many variations of ROUGE, including seven different ways of counting n-grams to measure the overlap between the generated and reference text. For example, ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 measure the overlap of unigrams, bigrams, trigrams, and four-grams respectively. ROUGE-L measures the longest sequence of words that match between the generated and reference summary. Finally ROUGE-S and ROUGE-SU use skip-bigrams and unigrams to calculate co-occurrence statistics, but these are less commonly used. The ROUGE metrics includes three sub-metrics: Recall (eq. 2.1), Precision (eq. 2.2) and F1 (eq. 2.3).

$$Recall = \frac{\textit{number of n-grams found in model and reference}}{\text{number of n-grams in }\textbf{reference}} \tag{2.1}$$

$$Precision = \frac{\textit{number of n-grams found in model and reference}}{\text{number of n-grams in }\textbf{model}} \tag{2.2}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.3}$$

Recall measures how many relevant words the model is able to capture, while precision considers whether the model is producing only relevant words and not including any unnecessary or irrelevant information. By combining these two factors, the F1 score provides a more reliable way than recall and precision to assess the overall performance of a model.

In line with all these findings, we use the ROUGE F1 to evaluate the performance of our approaches. The py-rouge package[1] is used to calculate these ROUGE scores.

2.4.2 Statistical Analysis in Evaluation

Statistical analysis is used in the evaluation of various systems, offering a systematic approach to compare different systems or methodologies. By applying statistical methods (Ross, 2010a), researchers can determine with greater confidence whether observed differences in performance are due to the model's characteristics or random variations in the data.

Hypothesis Testing

Hypothesis testing is a statistical method used to infer the validity of a hypothesis about a dataset, based on sample data (Ross, 2010b). It is an essential tool in statistical analysis for determining whether to accept or reject a null hypothesis (H_0) about the population. The null hypothesis typically represents a general or default position, such as no difference between two groups or no effect of a particular treatment. The alternative hypothesis (H_1) is the hypothesis contrary to the null hypothesis, often representing a new theory or the effect of an intervention.

Commonly used statistical tests in hypothesis testing include the z-test, t-test, ANOVA, and chi-square test, among others (Lehmann et al., 1986). The choice of test depends on the data distribution, sample size, and the number of samples being compared. Once the test is performed, a p-value is calculated, which indicates the probability of observing the test results under the null hypothesis. If the p-value is less than the chosen significance

[1] https://pypi.org/project/py-rouge/

level (usually 0.05), the null hypothesis is rejected, suggesting that the observed data are inconsistent with the assumption of no effect or no difference, thereby lending support to the alternative hypothesis.

In the evaluation of summarisation, hypothesis testing is employed to ascertain whether the differences in performance metrics, such as ROUGE scores, between various summarisation models are statistically significant or attributable to random chance. Researchers might set up a null hypothesis stating no difference between the performance of two summarisation methods and an alternative hypothesis indicating a significant difference. However, the subjective nature and high variability inherent in text summarisation tasks can limit the applicability and interpretation of these statistical tests. There is very limited research applying hypothesis testing in summarisation evaluation (Owczarzak et al., 2012; Steen & Markert, 2021), indicating a need for further exploration and validation in this area.

The appropriate statistical tests to compare model performance measured by ROUGE statistics for a dataset are, depending on the specific question, either Friedman's ANOVA or paired tests. However, in this thesis the summary performance statistics for comparator models often come from other published papers so the document specific ROUGE scores are not available for them. This means that the distributions of the performance metrics across documents cannot be plotted for comparator models, and that the covariances of the ROUGE scores between models that are required for these statistical tests to be applied cannot be calculated. Therefore, an indicative measure of the rough difference in ROUGE scores that is required for the difference to be unlikely due to chance alone is developed based on the unpaired t-test. The p-value from this test depends on the sample size of the dataset the models are applied to and the standard deviation of the ROUGE scores across documents (which cannot be calculated for comparator models for the same reasons as outlined above). For the NoWRank Sentence_Feature model (developed in chapter 5) the document specific ROUGE scores had a standard deviation of 16.4 for both the PuBMed and CNN datasets which have sample sizes of approximately 20,000 and 35,000 respectively. Three thresholds for statistical significance are considered for illustration: 5%, 1% and a 10 comparison Bonferoni corrected test at the 5% level. For the PubMed dataset the difference required in the mean ROUGE score to obtain a statistically significant result at these 3 levels is 0.32, 0.42, and 0.46 respectively. For the CNN dataset they are 0.25, 0.32, and 0.35 respectively. Moving forward we assume that a difference in ROUGE score of at least 0.5 (on the ROUGE 0-100 scale) and 0.005 (on the ROUGE 0-1 scale) to mean that the observed difference in performance for this metric is unlikely due to chance alone. This is reflected in our interpretation of results.

It is also important to note that hypothesis tests only consider how likely differences in observed model performance for a dataset are to be due to chance, they tell us nothing about any real differences in performance are likely sufficient to make a material difference when the models are employed in practice.

2.5 Conclusion

In this chapter, we conducted a literature review of previous research on topics related to our study. We began by examining the various areas of text summarisation, and then proceeded to investigate unsupervised methods, as well as clustering and graph techniques, which are relevant to our research on text summarisation. In the following chapters, we will introduce our proposed methods and present the results of our experimental evaluations.

Chapter 3

Sentence Similarity Measurement in Long Document Summarisation

In text summarisation, determining sentence similarity is one of the essential phases in producing a condensed version of a text. The process of measuring similarity directly impacts the removal of redundancy in the summarised text (Section 1.3.1 and Section 2.2.3). This chapter presents an enhanced method for computing sentence similarity, using Sentence-BERT models, aiming to improve long document summarisation performance. The chapter follows the subsequent format: Section 3.1 introduces the sentence similarity problem, and Section 3.2 elaborates on the specifics of our proposed method. In Sections 3.3 and 3.4, we present our experimental methodology and results. Finally, Sections 3.5 and 3.6 offer a discussion and conclusion for our work.

The work described in this chapter was published in the paper Extractive Financial Narrative Summarisation using SentenceBERT Based Clustering in Proceedings of the 3rd Financial Narrative Processing Workshop, Association for Computational Linguistics (ACL2021) (Gokhan et al., 2021).

3.1 Introduction

In text summarisation, similarity measurement plays an important role in producing a concise version of the original text by identifying and removing redundant information. By measuring the similarity between sentences or textual units, the summarisation system can identify those sentences that convey similar or overlapping information and eliminate unnecessary sentences. This helps to create a summary that accurately captures the main ideas and key information of the original text while maintaining its coherence and meaning. Therefore, in this study, we focus on sentence similarity.

Techniques such as Bag of Words (BoW) and the Term Frequency-Inverse Document Frequency (TF-IDF) are often used to represent text as real value vectors to aid in the calculation of semantic similarity. However, these techniques do not consider the fact that words can have different meanings and that different words can be used to represent similar concepts (Chandrasekaran & Mago, 2021). For example, consider two sentences: "George and Arthur studied English Literature and History." and "George studied English Literature and Arthur studied History." Although these two sentences have the same words, they have different meanings. Similarly, the sentences "Mary is allergic to dairy products." and "Mary is lactose intolerant." convey the same meaning, but they do not have the same set of words. These methods capture the lexical features of the text and are simple to implement, but they ignore the semantic and syntactic properties of the text. To address these drawbacks of lexical measures, various vector space models based on semantic similarity techniques have proposed over the last few decades (Section 2.2.3).

Sentence embeddings created with pre-trained language models provide vector representations of sentences wherein these vectors retain the underlying linguistic relationship between the sentences. Due to the fact that BERT (Devlin et al., 2019) achieves breakthrough performance in multiple NLP tasks, we focus on the SentenceBERT (Reimers &

Gurevych, 2019) model for sentence similarity problems in long document summarisation. To examine the SentenceBERT models' effect on summarisation performance, we participated in *The Financial Narrative Summarisation Shared Task for 2021* (FNS-2021) (El-Haj et al., 2021) and proposed a clustering-based summarisation method. We applied this method to a dataset of financial documents consisting of 80 pages on average.

FNS-2021 aims to evaluate the performance of automatic summarisation methods applied to annual reports from UK corporations listed on The London Stock Exchange (El-Haj et al., 2020). Compared to reports prepared by US companies, these reports have a notably less rigid structure that makes summarisation particularly challenging. These reports can be divided into two main sections. The first section is a "narrative" section which is also known as a "front-end" section containing textual information and reviews by the company's management and board of directors; the second section is the "back-end" section which contains financial statements that tend to consist of tables of numerical data. The FNS-2021 shared task entails determining which essential narrative sections are and then summarising these to achieve a summary of approximately 1000 words.

This chapter will discuss the solution we developed for the FNS-2021 shared task. The data set is the annual reports in the financial field provided by the organiser. Our approach to the FNS-2021 shared task involves an initial analysis of the documents, from which an intermediate set of custom documents is generated, consisting of the most important sections of the reports. Using SentenceBERT, we create vector representations for this intermediate document set. Finally, we cluster the vectors and select sentences from each cluster to generate the final report summaries. The results obtained provide evidence for the efficacy of our proposed method.

3.2 Methodology

In this section, we introduce our cluster-based approaches to long document summarisation using SentenceBERT models. Figure 3.1 displays the flow diagram of our approach that we explained in this chapter.

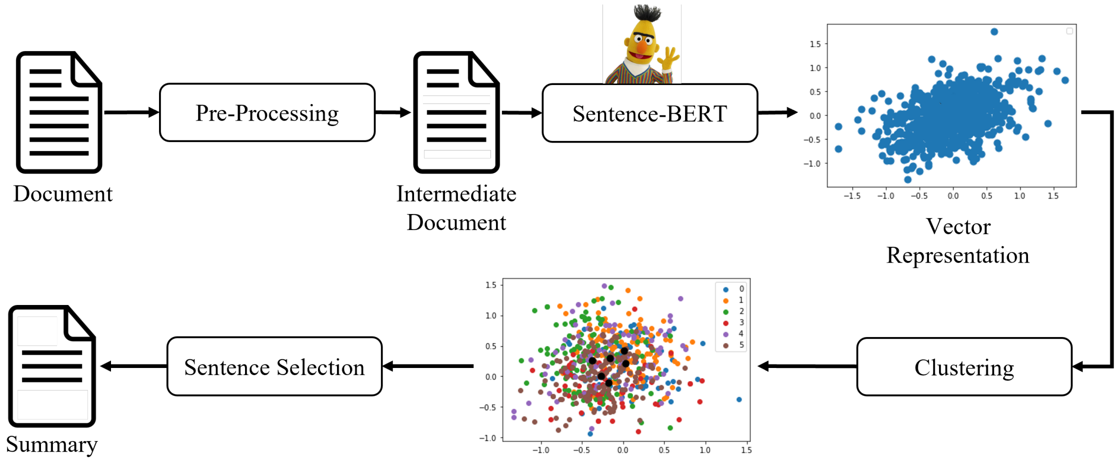


Figure 3.1: Our cluster-based summarisation system pipeline.

3.2.1 Financial Narrative Summarisation Dataset

For this shared task, the data consist of 3,863 United Kingdom annual reports for corporations listed on the London Stock Exchange (LSE) between 2002 and 2017. Annual reports in the UK are long papers that average approximately 80 pages, some exceeding 250 pages. For the FNS-2021 shared task, these annual reports are separated into three sections: training, testing, and validation (Table 3.1)

Table 3.1: FNS-2021 shared task dataset.

Data Type	Training	Validation	Testing
Report Full Text	3000	363	500
Gold Summaries	9873	1250	1673

The complete text of each yearly report, as well as the gold-standard summaries, are included in the training and validation sets. Each annual report has at least three gold-standard summaries on average, with some reports including up to seven gold-standard summaries. The task participants are only provided access to the full texts for the testing data set.

3.2.2 Pre-Processing on Financial Documents

Due to the natural structure of the annual reports and the conversion of the dataset from PDF files, sentences and structures that disrupt the meaning integrity should be removed from the documents. Therefore, narrative sections are defined by applying basic pre-processing operations to the whole dataset. The processes applied are explained below:

- **Cleaning of Parentheses:** All parentheses throughout the document have been cleaned. Structures in parentheses are often used for explanation. For this reason, this cleaning process is widely used among text summarisation pre-processes.
- **Cleaning Non-Alphanumeric Characters and Digits:** The numbers of non-alphanumeric characters and the number of digit characters in the sentences are calculated. Our evaluations based on the ROUGE metric determine that sentences containing more than 10% of these characters lack meaningful content, as they primarily comprise table or formula transformations. The sentences with this structure are cleared from the document.
- **Clearing Short and Long Sentences:** As a result of our evaluations, it is observed that if the sentence contains fewer than seven words or more than eighty words, it does not produce meaningful results for the summary created. For this reason, these sentences have identified and cleared from the document.

- **Section Segmentation:** Section segmentation plays a crucial role in financial document summary systems. In these documents, it is necessary to create purified, clean intermediate documents that do not include sections that are not expected to be included in the summary. The use of such intermediate documents saves time and enables the production of higher-quality summaries. Hence, the gold standard summaries are first examined in detail. In the provided training set, we find that the main narrative sections are mostly under four headings: "Chief Executive's review", "At a glance", "Highlights", and "Chairman's Statement". These sections typically include summarised financial topics. The other sections contain statistical data, tables, graphs, and diagrams. Therefore they are not included in the organiser's gold summary. The focus of the study is to determine narrative segments to build our summarisation system. Therefore, a condensed intermediate document is created to cover only the sections we expected to appear in the final summaries.

The ROUGE is used to measure the effect of each process in the cleaning phase on the summary quality. The effect of each change on the overall results is calculated by making variations. These evaluations showed that title cleaning and Post of Tagging operations did not have an effect on the results. These transactions are not included in preprocessing stage since they create a transaction load. Table 3.2 shows the average number of sentences and words of the documents obtained after all pre-processing was completed.

Table 3.2: The average counts of sentences and words on the FNS Dataset after the pre-processing stage.

	# **Sentences**	# **Words**
Original Document	2059	76050
After Pre-processing	1823	51785

3.2.3 Pre-Trained Language Models with SentenceBERT

Pretrained language models have become increasingly popular for NLP tasks. GBT, BERT, and RoBERTa, are among the most widely used pre-trained language models and have demonstrated significant improvements in various language tasks compared to previous methods. These models build on the concept of word embeddings and are trained to learn contextual representations from large- scale corpora. By grasping the meaning and context of words, these models can generate high-quality sentence embeddings that are beneficial for NLP (Y. Liu & Lapata, 2019).

Compared to the word-level vectors produced by traditional methods, BERT has the capacity to train sentence-level vectors and extracts more information from the context. The sentence-level vector is more suitable for the use of downstream NLP tasks (L. Zhao et al., 2020). The BERT architecture is chosen due to its high performance over other NLP algorithms for Sentence Embedding. BERT is built on transformer architecture. However, the targets can be privatised with pre-training. First, randomly masks 10% to 15% of the words in the training data and tries to predict these masked words. Then, by taking an input sentence and a candidate sentence, predict whether the candidate sentence correctly follows the introductory sentence. Training this process can take several days, even with a significant amount of GPU. Google published two BERT models for public consumption, one containing 110 million and the other 340 million parameters (Devlin et al., 2019).

In our studies, the core BERT implementation uses the $pytorch-pretrained-BERT$ library (Wolf et al., 2020). At its core, the library is a Pytorch wrapper around Google's pre-trained implementations of the models. On top of the original BERT model, the pytorch-pretrained-BERT library also contains the OpenAi GPT-2 model, which is a network that expands on the original BERT architecture. Table 3.3 contains a brief presentation of the most critical models which we examine in our study.

Table 3.3: List of pre-trained models used in our experiments.

Architecture	**Shortcut name**	**Details of the model**
BERT	bert-base-uncased	12-layer, 768-hidden, 12-heads, 110M parameters. Trained on lower-cased English text.
XLNet	xlnet-base-cased	12-layer, 768-hidden, 12-heads, 110M parameters. XLNet English model
DistilBERT	distilbert-base-uncased	6-layer, 768-hidden, 12-heads, 66M parameters. bert-base-uncased based
ALBERT	albert-base-v1	12 repeating layers, 128 embedding, 768-hidden, 12-heads, 11M parameters.
RoBERTa	roberta-large	24-layer, 1024-hidden, 16-heads, 355M parameters. BERT-large based
BERT	bert-base-nli-mean-tokens	Base Model:bert-base-uncased Pooling:Mean Pooling Training Data:NLI (Conneau et al., 2017)
BERT	bert-base-nli-stsb-mean-tokens	Base Model:bert-base-uncased Pooling:Mean Pooling Training Data:NLI+STSb(Cer et al., 2017)
RoBERTa	roberta-large-nli-mean-tokens	Base Model:roberta-large Pooling:Mean Pooling Training Data:NLI (Conneau et al., 2017)
RoBERTa	roberta-large-nli-stsb-mean-tokens	Base Model:roberta-large Pooling:Mean Pooling Training Data:NLI+STSb(Cer et al., 2017)

SentenceBERT is a modification of the pre-trained BERT network that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SentenceBERT while maintaining the accuracy from BERT (Reimers & Gurevych, 2019). Due to its performance in the larger pre-trained BERT model, SentenceBERT is ultimately chosen for our method. In our studies, the sentences are vectorised with SentenceBERT. Different models are used in the vectorisation phase. These models and their performance are presented in Section 3.3.

3.2.4 Clustering with K-Means

Clustering is a process which partition a set of data into a set of meaningful sub-classes called clusters. For developing our method, the **Scikit-Learn Clustering**[1] library was used. The clustering algorithms available in the Scikit-Learn library are shown in Table 3.4.

Table 3.4: A comparison of the Scikit-Learn library's clustering algorithms.

Method name	**Parameters**	**Scalability**	**Use case**
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry
Ward hierarchical clustering	number of clusters or distance threshold	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints
Agglomerative clustering	number of clusters or distance threshold, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non Euclidean distances
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation

As indicated in Table 3.4, K-Means excels in handling "very large n_samples and medium n_clusters with MiniBatch code," underscoring its capability to manage substantial datasets, making it well-suited for summarising long documents. Its flexibility in configuring the number of clusters provides customisable levels of summarization, facilitating either comprehensive or concise document summaries according to specific needs. For these reasons, K-Means is chosen in this study to cluster sentence embeddings derived from the BERT model.

[1] https://scikit-learn.org/stable/modules/clustering.html

Let $X = \{x_1, ..., x_n\}$ be a data set in a d-dimensional Euclidean space R^d, Let $A = \{a_1, ..., a_c\}$ be c cluster centers. Let $z = [z_{ik}]_{nxc}$, where z_{ik} is a binary variable (i.e. $z_{ik} \epsilon \{0, 1\}$) indicating if the data point x_i belongs to $k - th$ cluster, $k = 1, ..., c$. The K-Means objective function is:

$$z_{ik} = \begin{cases} 1 & if||x_i - a_k|| = min_{1 \leq k \leq c}||x_i - a_k||^2 \\ 0 & otherwise \end{cases} \tag{3.1}$$

$$a_k = \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} z_{ik}} \tag{3.2}$$

where $||x_i - a_k||$ is the Euclidean distance between the data point x_i and the cluster center a_k (Likas et al., 2003).

The idea of clustering sentences in a high-dimensional area has also been used in the past for ATS (Bookstein et al., 1995; Maybury, 1995; McKeown et al., 1999). In these methods, ATS identifies the most central and important sentences in a cluster such that they cover the critical information related to the cluster's main subject.

Algorithm 1 Pseudo-code of the K-Means algorithm

```
Specify the number k of clusters to assign
Randomly initialise k centroids
repeat
  expectation: Assign each point to its closest centroid
  maximisation: Compute the new centroid of each cluster
until The centroid positions do not change
```

The K-Means clustering algorithm is seen in Algorithm 1. When this algorithm is adapted for sentence-based summarisation systems, the cluster number is associated with the number of sentences that should be included in the summary. Points represent sentence vectors are created using various methods. In the literature, cluster-based summarisation systems generally use TF-IDF, BoW, SVD, word2vec, or GloVe. In our method, we use SentenceBERT models for cluster-based unsupervised summarisation. The sentences are clustered after creating sentence embeddings with SentenceBERT models. The number of clusters is assigned as the number of sentences that should be included in the summary. Each cluster contains sentences that are semantically similar to each other. The summary is completed by choosing the closest point (a.k.a sentence) to the centroid from each cluster.

3.3 Experimental Study

In this section, we explain the experimental setup to evaluate the performance of the proposed model.

3.3.1 Pre-processing

In pre-processing steps, the python **re**[2] package is used for regular expressions. It has been observed that the "Chief Executive's review", "At a glance", "Highlights", and "Chairman's Statement" sections are the sections that should be included in the summary. These sections are generally in the first 10% of each report. Therefore, we extract these and create a new data set containing only these sections. On this new condensed dataset, sentences are tokenised using the **tokenise**[3] package from the **NLTK**[4] library (Bird & Loper, 2004).

[2] https://docs.python.org/3/library/re.html

[3] https://docs.python.org/3/library/tokenize.html

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

Table 3.5: ROUGE results of pre-trained models applied on 30 documents.

	Summariser				**TransformerSummariser**				**SentenceTransformer**			
	bert-base-uncased	xlnet-base-cased	distil-bert-base-uncased	albert-base-v1	bert-base-uncased	xlnet-base-cased	distil-bert-base-uncased	albert-base-v1	bert-base-nli-stsb-mean-tokens	bert-base-nli-mean-tokens	roberta-large-nli-mean-tokens	roberta-large-nli-stsb-mean-tokens
Time(min)	72	60	43	49	40	44	30	48	93	45	41	37
Avg.Sent.	1709	1709	1709	1709	1709	1709	1709	1709	1709	1709	1709	1709
Avg.Word	60796	60796	60796	60796	60796	60796	60796	60796	60796	60796	60796	60796
Sum.Avg. Sent.	34	35	32	31	34	35	32	31	22	27	28	29
Sum.Avg. Word	954	956	961	963	949	953	950	966	932	939	947	908
R1 (P)	42.87	43.43	42.95	43.52	42.03	43.09	43.04	43.52	43.83	44.38	**44.39**	43.37
R1 (R)	48.14	48.52	48.01	48.39	47.41	48.39	48.52	48.33	50.2	50.57	50.23	**51.09**
R1 (F1)	44.98	45.46	45	45.51	44.24	45.24	45.24	45.44	46.41	**46.88**	46.76	46.56
R2 (P)	11.48	11.36	11.09	12.35	10.5	11.52	11.58	12.28	12.57	**13.01**	12.45	12.14
R2 (R)	12.89	12.54	12.28	13.62	11.8	12.82	12.9	13.57	14.32	**14.73**	14.15	14.33
R2 (F1)	12.04	11.8	11.55	12.85	11.02	12.03	12.08	12.79	13.27	**13.68**	13.14	13.05
R3 (P)	4.99	4.62	4.35	5.55	4.07	4.81	4.86	5.48	6.02	**6.14**	5.76	5.75
R3 (R)	5.64	5.07	4.84	6.1	4.6	5.33	5.43	6.07	6.87	**6.98**	6.64	6.79
R3 (F1)	5.25	4.78	4.55	5.77	4.29	5.01	5.08	5.72	6.37	**6.47**	6.13	6.18
R4(P)	3.51	3	2.83	4.03	2.72	3.19	3.29	3.97	4.28	**4.29**	4.08	4.13
R4 (R)	4.01	3.32	3.17	4.45	3.09	3.55	3.73	4.42	**4.92**	4.9	4.75	**4.92**
R4 (F1)	3.72	3.12	2.97	4.2	2.88	3.34	3.48	4.16	**4.55**	4.53	4.37	4.47
RL (P)	34.71	35.05	34.76	35.21	34.29	34.89	35.02	35.22	35.57	35.91	**36.57**	35.55
RL (R)	38.25	38.38	38.05	38.45	37.91	38.33	38.71	38.41	39.74	40.06	40.53	**40.72**
RL (F1)	36.18	36.41	36.12	36.58	35.82	36.32	36.56	36.54	37.3	37.65	**38.23**	37.75
RW (P)	13.29	13.3	13.07	13.44	12.92	13.21	13.29	13.39	13.64	13.74	**13.97**	13.57
RW (R)	3.84	3.8	3.73	3.82	3.75	3.79	3.85	3.81	4.01	4.04	4.07	**4.14**
RW (F1)	5.92	5.88	5.78	5.92	5.78	5.86	5.94	5.9	6.17	6.21	6.28	**6.31**

3.3.2 Sentence Embedding

Pre-trained models that can be used for sentence embedding are determined. After determining the narrative sections with the pre-processing operations applied, the models shown in Table 3.3 are applied to the dataset.

The results are shown in Table 3.5. The same 30 documents are used when creating experimental results. While determining the sample documents, care has been taken to ensure that they are in different sizes to provide diversity. When we evaluate the experimental results, it is observed that the SentenceTransformer (Reimers & Gurevych, 2019) based models were more successful due to both shorter processing times and higher results. After conducting the analysis, we use three models from SentenceTransformer that were specifically designed for clustering or semantic search purposes. These models include *nli-mpnet-base-v2*[5] , *distiluse-base-multilingual-cased-v2*[6], *nli-distilroberta-base-v2*[7].

3.3.3 Clustering

Sentence embeddings, when derived from SentenceBERT models, are clustered using the K-Means clustering algorithm. In alignment with the shared task's guidelines, summaries are required to be no more than 1000 words in length. Accordingly, the K value—which is reflective of the number of sentences in the summary and analogous to the number of clusters—must exhibit inherent flexibility.

To ascertain the distribution of word counts across sentences for the definition of the K value, an empirical analysis is undertaken. This analysis focuses on determining the necessary number of sentences for summaries that are constrained to 1000 words. The results

[5] https://huggingface.co/sentence-transformers/nli-mpnet-base-v2

[6] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

[7] https://huggingface.co/sentence-transformers/nli-distilroberta-base-v2

are exhibited in Figure 3.2, showing the distribution of sentence numbers per 1000 words across a selection of documents.

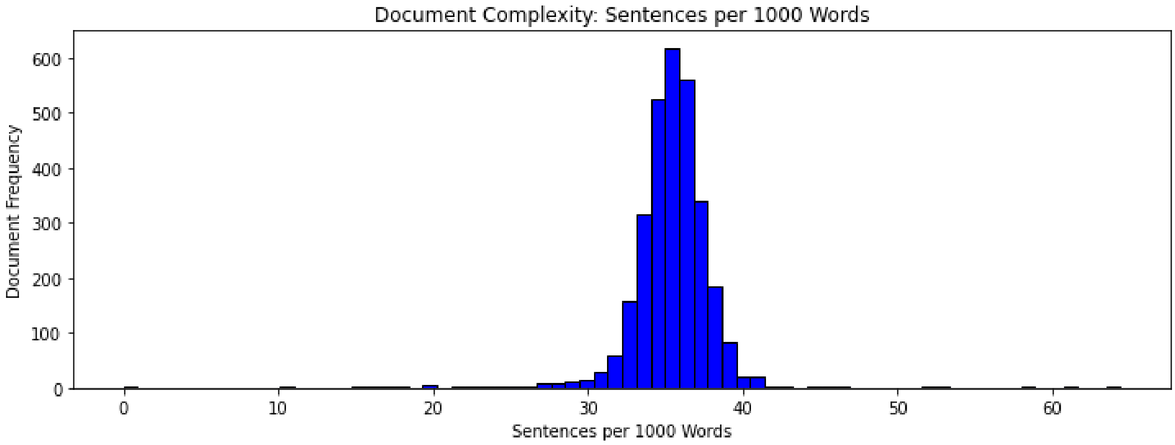


Figure 3.2: The distribution of the number of sentences per 1000 words across a collection of documents.

Insights from Figure 3.2 are deemed critical for this calibration: to ensure that summaries remain within the 1000-word limit, the K value is set to smaller than 32 , contingent upon each document's length and complexity. This correlation is instrumental in customising the length of summaries to the established word count limit.

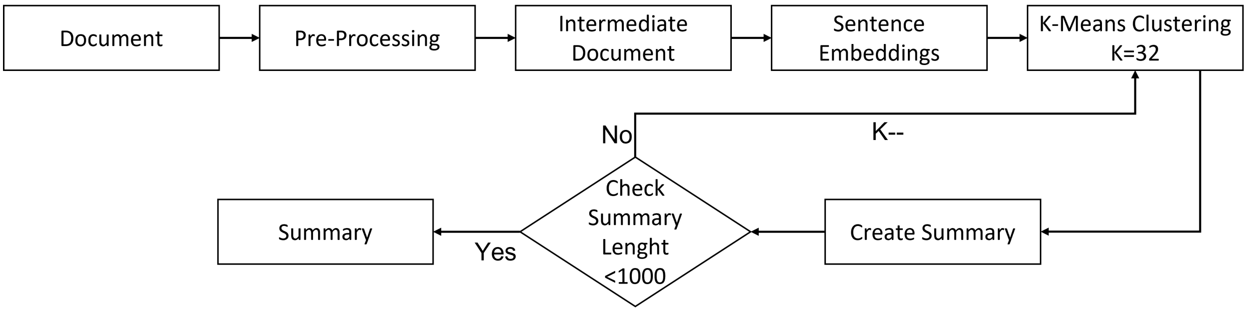


Figure 3.3: Our cluster-based summarisation system pipeline with optimum cluster number for 1000-word summaries.

Therefore, the proposed method presented in Figure 3.1 has been updated to Figure 3.3 to determine the optimal number of clusters for generating a 1000-word summary.

As a result of implementing these steps, summaries consisting of an average of 27 sentences and 945 words were created from a dataset of 500 documents, which originally contained an average of 2059 sentences and 76050 words.

3.4 Results

Following the development of the aforementioned system, we evaluate our method as follows. The FNS-2021 Shared Task contest decides to use the ROUGE2 package[8] to evaluate the system outputs. The ROUGE2 package is a multilingual tool that implements ROUGE (Lin, 2004) metrics. The performance of our method is evaluated with ROUGE-1, ROUGE-2, ROUGE-SU4 and ROUGE-L. Since the organisers of the task did not release the golden summaries of the test dataset, we performed the evaluation on the validation dataset. Table 3.6 shows the results for the three highest-performing models.

Table 3.6: ROUGE results of our experiment on the FNS2021 shared task validation dataset.

	Model Name	**R-1/F**	**R-2/F**	**R-L/F**	**R-SU4/F**
Our System 1	nli-mpnet-base-v2	0.48	0.25	0.41	0.29
Our System 2	distiluse-base-multilingual-cased-v2	0.48	0.25	0.40	0.29
Our System 3	nli-distilroberta-base-v2	0.47	0.25	0.40	0.29

The organisers on the test set measure the performance of the generated summaries. Table 3.7 shows the organisers' calculated scores for the three systems we provide. In Table 3.7, the results of our systems, the results of the baseline TEXRANK (Mihalcea & Tarau, 2004) and LEXRANK (Erkan & Radev, 2004) algorithms, the results of PointT-5 (Singh, 2020), SumTO (La Quatra & Cagliero, 2020) and HULAT (Baldeon Suarez et al., 2020),

[8] https://github.com/kavgan/ROUGE-2.0

which are the systems with the highest performance in the FNS-2020 Shared task, and the results of the topline MUSE (Litvak et al., 2010) algorithm are presented.

Table 3.7: ROUGE results measured by the FNS2021 shared task organisers on the test dataset.

System	**R-1/ F**	**R-2/ F**	**R-L /F**	**R-SU4 /F**
TEXTRANK (baseline)	0.17	0.07	0.21	0.08
LEXRANK (baseline)	0.26	0.12	0.22	0.14
PointT-5	0.46	0.28	0.45	0.28
SumTO	0.42	0.24	0.39	0.26
HULAT	0.44	0.26	0.38	0.26
MUSE (topline)	0.50	0.28	0.45	0.32
Our System 3-1	0.47	0.25	0.40	0.29
Our System 2	0.48	0.26	0.40	0.29

The results, as shown in Table 3.7, indicate that the MUSE system, designated as the topline, achieves the highest scores across most metrics, with an R-1/F score of 0.50, R-2/F of 0.28, R-L/F of 0.45, and R-SU4/F of 0.32. Our systems demonstrate lower, but competitive performance with R-1/F scores of 0.47 and 0.48, R-2/F scores of 0.25 and 0.26, R-L/F scores of 0.40 for both, and R-SU4/F scores of 0.29 for both, respectively. Notably, our systems, like the baseline systems, are developed using unsupervised approaches, yet they show a significant improvement in performance compared to TEXTRANK and LEXRANK. This underscores the effectiveness of our proposed unsupervised summarization techniques, highlighting their capability to achieve competitive results without the need for a training dataset, a common requirement for supervised systems such as MUSE, which reports the highest ROUGE scores.

3.5 Discussion

This study proposes a SentenceBERT-based clustering approach as an unsupervised method for the FNS Shared task. As a result of this approach, extractive summaries of less than 1000 words are created. In order to create high-quality summaries in this dataset, first and foremost, it is necessary to define the "Chief Executive's review", "At a glance", "Highlights", and "Chairman's Statement" sections that form the basis of gold summaries. However, defining this section makes this task difficult because the documents are produced as a result of converting the PDF documents. For this reason, the pre-processing phase is extended in our work.

Another challenge in this task is producing 1000-word summaries. The basis of our proposed approach is clustering. In order to create summaries of 1000 words, we need to limit the number of cluster sets to a minimum of 25. However, it is necessary to determine the ideal number of clusters according to the data distribution in clustering approaches. This number of clusters varies depending on the documents, and the restriction of the number of clusters causes sentences with different meanings to be included in the same cluster.

Our approach demonstrates the applicability of SentenceBERT models to measure sentence similarity in long document summarisation. However, one of the most significant disadvantages of our proposed clustering method is that it only offers a similarity-based approach. Although the sentence similarity is substantial in ATS, more is needed. It is argued that the summary quality will increase with a system that includes more information about sentences.

3.6 Conclusion

In this chapter, we presented an enhanced method to measure sentence similarity for long document summarisation. For showing effectiveness of sentence similarity in ATS, we describe an extractive summarisation approach. The proposed approach relies on clustering sentence vectors created with sentence embedding. First, an intermediate document dataset covering the most important parts of the documents is prepared. Then, pre-trained language representation model Bidirectional Encoder Representations from Transformers (BERT) is utilised to generate sentence embeddings. Finally, the K-means clustering algorithm is applied to find similar sentences and a sentence vector representing the set is selected from each cluster for the final summary. Three systems are created using different sentence embedding models are submitted. The performance of the obtained summaries is measured with the ROUGE metric. Our approach outperforms the baseline algorithms in terms of performance when is compared to the literature, whereas the topline algorithm produce partially near results.

The study shows SentenceBERT's efficacy in measuring sentence similarity, but concluded that clustering-based methods are inadequate for long document summarisation. Therefore, the focus of the study is shifted to graph-based methods. The outcomes gained from this study are applied to the development of graph-based methods for overcome the existing limitations of graph-based methods. The next chapter presents our novel document representation graph model proposed to mitigate the limitations of graph-based summarisation systems.

Chapter 4

Node-Weighted Graph Models for Document Summarisation

In this chapter, we present a graph-based unsupervised extractive summarisation approach based on a novel node-weighted graph model for document representation Section 4.1 introduces the problem, and Section 4.2 elaborates on the specifics of our proposed method. In Sections 4.3 and 4.4, we present our experimental methodology and results. Finally, Sections 4.5 and 4.6 offer a discussion and conclusion for our work.

The work described in this chapter was published in the paper GUSUM: Graph-Based Unsupervised Summarisation Using Sentence Features Scoring and SentenceBERT in Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, The 29th International Conference on Computational Linguistics (COLING 2022) (Gokhan et al., 2022).

4.1 Introduction

Unsupervised graph-based methods define sentences as nodes and the edges between the nodes are weighted based on the similarity of sentences (Erkan & Radev, 2004; Liang et al., 2021; J. Liu et al., 2021; Mihalcea & Tarau, 2004; S. Xu et al., 2020; Zheng & Lapata, 2019). Summarisation is therefore a node selection task in which nodes (i.e. sentences) semantically relevant to other nodes are chosen for placement in the final summary.

Node weights play a fundamental role in the functioning of graph-based approaches. The weights of the nodes provide an essential quantitative measure of their significance within the graph. These weights inform subsequent analytical operations, such as node selection or ranking, that are essential for producing meaningful outputs such as summaries or identifying key information. Therefore, the specification of node weights is a critical prerequisite for the successful application of graph-based approaches. Current methodologies for graph-based text summarisation presume that node weights are unique, or they are calculated based on the weights of the edges between the nodes (Section 2.3.2). While these approaches have been successful in producing summaries that capture the key information from the input text, they still need improvement.

Recent research in other disciplines has shown that full-weight graphs, in which nodes are assigned explicit values that reflect their relevance to the research field, can improve the accuracy of systems in graph analysis (Singh et al., 2020). Therefore, customising the weights assigned to nodes in graph-based summarisation systems is a potential approach to improving their performance in generating high-quality summaries. By assigning more weight to the nodes that are most relevant to the document's content and reducing the weight of less relevant nodes, the resulting summary is likely to be more informative and concise. Additionally, by incorporating fully weighted graphs into the summarisation process, graph-based summarisation systems can leverage the full spectrum of information available in the input text, leading to better summarisation outcomes. Therefore, the customisation of node weights in graph-based text summarisation systems has the potential to enhance their effectiveness and accuracy in producing high-quality summaries.

In this chapter, we present a novel node-weighted graph model to improve performance of unsupervised extractive summarisation. To show the effect of node weight in ATS, we develop a **G**raph-Based **U**nsupervised **Sum**marisation(GUSUM) method. We first define the sentence feature scores, represented by the vertices, which indicate the importance of each sentence in the document. After this stage, we use SentenceBERT to obtain sentence embeddings to capture the sentence's meaning better. In this way, we define the edges of a graph to represent semantic similarities between sentences. Next, we create an undirected graph that includes sentence significance and similarities between sentences. In the last stage, we propose a ranking algorithm to determine the most important sentences in the document from the graph created. Experiments on CNN/Daily Mail, New York Times, arXiv, and PubMed datasets show that our approach achieves high performance for unsupervised graph-based summarisation when evaluated against other automated text summarisations and by humans.

4.2 Methodology

In this section, we describe our unsupervised summarisation method GUSUM. The system is composed of four main steps: first, we calculate sentence features for defining node weights; secondly, we produce sentence embeddings using SentenceBERT to measure sentence similarities; thirdly, we create a graph by comparing all the pairs of sentence embeddings obtained; finally, we rank the sentences by their degree centrality in this graph. Figure 4.1 gives an overview of the whole proposed method.

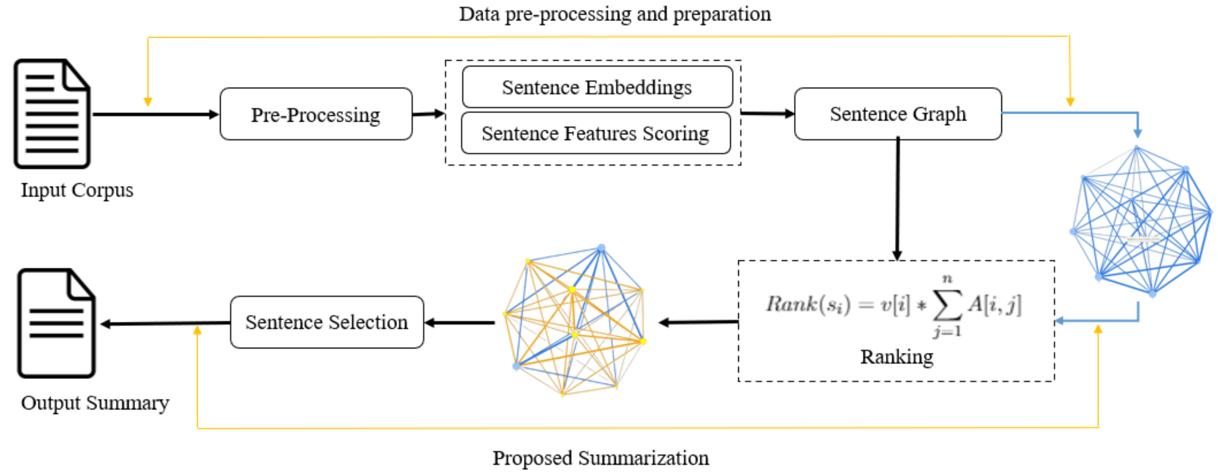


Figure 4.1: Our graph-based summarisation system pipeline using our node-weighted graph model.

4.2.1 Node-Weighted Graph Model

In graph-based methods, the nodes in a graph represent the data points or entities, and the edges represent the relationships or connections between them. The importance of a node weight in a graph model is that it represents the significance or relevance of a particular node to the problem at hand. For example, in a social network graph, a node representing a high-influence user would likely have a higher weight than one representing a low-influence user. This weight can be used in various graph-based algorithms such as centrality measures, which determine the importance of a node within the graph, or in recommendation systems, where the weight can be used to determine the relevance of a particular node to a given user. As in the example of the social network, some sentences or phrases have a low effect on the summary, while the effect of some contains information that must be included in the summary. In the context of text summarisation, the effect of specific sentences or phrases on the summary has been overlooked in current graph-based systems. In this study, we posit that node weights are a crucial component in graph-based ranking systems and propose a text summarisation model that incorporates them.

4.2.2 Computing Sentence Features for Node Weights

The initial stage in our proposed pipeline involves the generation of node weights from the input documents. Conventionally, in embedding-based systems, sentence features are transformed into dense vector representations, which are then used as attributes to represent the data for the task at hand (Suanmali, Binwahlan, & Salim, 2009). Unlike traditional methods, GUSUM uses sentence features to determine the initial values of the nodes in the generated graph.

To identify the appropriate sentence features, GUSUM focuses on four key aspects of each sentence: sentence length, sentence position, ratio of proper nouns to tokens and ratio of numerical tokens to token (Shirwandkar & Kulkarni, 2018). It is worth noting that while there may be room for diversification in terms of the selection of features, the focal point of our approach is on the impact of node weights on graph-based methods. Hence, the selection of these four features is strategically aimed at presenting the impact of node weights on graph-based text summarisation methods.

Sentence length: This feature is useful for filtering out short phrases commonly found in news articles, such as dates and author names. Short sentences do not contain much information and are not expected to belong in the summary. To find the important sentence based on its length, the feature score is calculated using 4.1:

$$Score_{f1}(S_i) = \frac{Number\ of\ Word\ in\ S_i}{Number\ Word\ in\ Longest\ Sentence} \tag{4.1}$$

Sentence position: The position of a sentence within a text can offer significant indications regarding its relevance to the entire document. In many cases, the first and the last sentence of a document are important and involve essential information that should be included in a summary. Position feature is calculated using 4.2:

$$Score_{f2}(S_i) = \begin{cases} 1 & \textit{if the first or last sentence} \\ \frac{N-P}{N} & \textit{if others} \end{cases} \tag{4.2}$$

where, N is the total number of sentences and P is the position of the sentence.

Proper nouns: Usually, sentences that contain more proper nouns are more important than those that contain fewer and hence more likely to be important for summarisation. The score for this feature is calculated as the ratio of the number of proper nouns in a sentence over the sentence length as shown in 4.3.

$$Score_{f3}(S_i) = \frac{\textit{Number of Proper Noun in } S_i}{\textit{Length } S_i} \tag{4.3}$$

Numerical tokens: In scientific or financial reports, numerical information can be crucial for understanding the main ideas and conclusions of the text. Statistical summarisation algorithms that take ratio of numerical token into account can increase the chance that the summary includes the most important numerical information and is calculated with 4.4:

$$Score_{f4}(S_i) = \frac{\textit{Number of } Numeric_i}{\textit{Length } S_i} \tag{4.4}$$

where, $num\_numeric_i$ is the total number of numerical tokens in sentence i.

Once the scores for each sentence are determined based on these four features, the sum of the scores is assigned the weight of the node that represents the respective sentence. This step ensures that the node weights are effectively incorporated into the generated graph, thus contributing to the overall efficacy of the approach.

Table 4.1: A 20-sentence sample document obtained from the CNN/DM dataset.

S1	Call it a little piece of heaven for a family torn apart by tragedy.
S2	Back in July, Sierra Sharry and Lane Smith were just about to become parents.
S3	Sharry was eight months pregnant.
S4	But then Smith fell and hit his head.
S5	He was taken to the OU Medical Center in Oklahoma City.
S6	Smith never recovered.
S7	"July 13th 2014 was the absolute worst day of my life," Sharry posted on Facebook.
S8	"I lost my best friend.
S9	The father of my unborn child."
S10	Their son Taos arrived a few weeks later.
S11	When it was time for his 6-month pictures, Sharry had a special request.
S12	Maybe the photographer could make their family complete, just for one picture .
S13	"They asked me if I would be willing to 'play around' with capturing their first family photo by editing Taos' daddy in one of their pictures, Kayli Rene' Photography posted on Facebook.'
S14	"I just got to thinking, we don't have a picture with Lane in it," the new mom told CNN affilaite KOCO.
S15	"one that has him looking over his family's shoulder."
S16	"Lane's not physically here with us, of course, but that picture represents to us that he is always watching over us and he will always be there for us no matter what," Sharry said.
S17	"The family photo has become a social media sensation after appearing on the photographer's Facebook page this week."
S18	It has some 193,000 likes and more than 24,000 shares.
S19	"I can't believe she actually did this," Sharry said.
S20	"It's like amazing and apparently everyone else thinks it is too."

Table 4.2: The gold summary of the 20-sentence sample document obtained from the CNN/DM dataset.

	Sierra Sharry was eight months pregnant when her son's father died .
	A photographer was able to add Lane Smith to the family photo .

Table 4.3: Node weights based on sentence feature scoring for 20-sentence sample document obtained from the CNN/DM dataset.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
2.365	**2.652**	2.0463	2.180	**2.509**	2.097	2.268	1.820	1.795	1.880	1.932	1.720	2.453	2.119	1.568	2.298	1.737	1.236	1.512	**2.365**

Table 4.1 shows a sample 20-sentence document obtained from the CNN/DM dataset and Table 4.3 shows the node weights of the sentences in the document. The scores, generated by employing the sum of the four statistical methods, suppose that Sentence 1, Sentence 2,

Sentence 5, and Sentence 20 are the highest significance in the given document (Table 4.3). A visual comparison of these results with the gold summary (Table 4.2) shows that the sentences selected only based on statistical methods are insufficient for a comprehensive and effective summary. However, we hypothesis that these methods are sufficient for assigning node weights.

4.2.3 Computing Sentence Embeddings for Edge Weights

The second stage of our pipeline is to generate sentence embeddings for measure sentence similarities. At this stage, we use SentenceBERT (Reimers & Gurevych, 2019) to create our sentence embeddings by applying the findings from the research we presented in Chapter 3. SentenceBERT[1] embeddings to represent sentences as fixed-size vectors. Thus, all sentences from the source document are mapped in the same semantic space and taken as inputs to the system.

4.2.4 Generation of the Sentence Graph

A given document D, it contains a set of sentences $(s_1, s_2, ..., s_n)$. Graph-based algorithms treats D as a graph $G = (V; E)$ (Figure 4.2). $V = (v_1, v_2, ..., v_n)$ is the node (vertex) set where v_i is the representation of sentence s_i. E is the edge set, which is an $n \times n$ matrix. Each $= e_{i,j} \in E$ denotes the weight between node v_i and v_j.

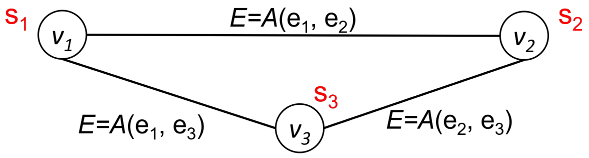


Figure 4.2: A simple, fully connected, undirected graph sample.

[1] https://www.sbert.net/

The first step to build the sentence graph is to generate the edges that represent semantic sentence similarities. Cosine similarity can be used as a measure to find similarity between sentences of the graph. In this step, all the pairwise Cosine similarities are gathered in a matrix. Cosine similarity is defined as:

$$Cosine\,Similarity = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} \mathrm{A}_i^2}\sqrt{\sum_{i=1}^{N} \mathrm{B}_i^2}} \tag{4.5}$$

where A_i and B_i are the components of vector A and B respectively.

Let $(e_1; e_2; ...; e_n)$ be a set of vectors , where e_i is the sentence embedding of s_i. Its edges are weighted according to the cosine similarities of the corresponding sentence embeddings. Next, we compute the matrix A with 4.6:

$$A[i, j] = Cosine\,Similarity(e_i; e_j) \tag{4.6}$$

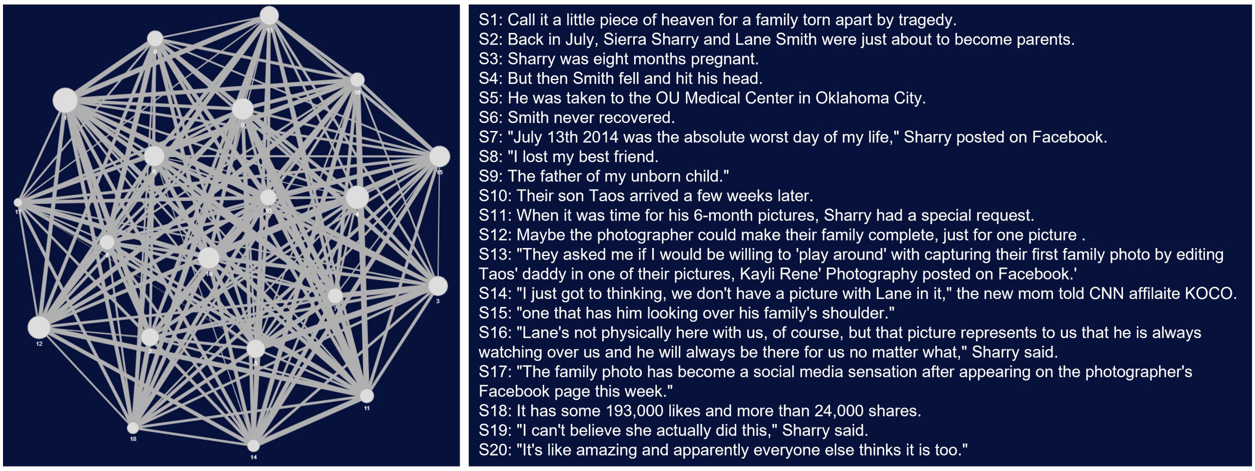
Thus, matrix A can be interpreted as the adjacency matrix of an undirected weighted complete graph. The adjacency matrix is assigned as the edge weights of the graph. An adjacency matrix of a 20-sentence document is showen in Table 4.4.

As a second step, node values are assigned by using sentence feature scores. $V = (v_1, v_2, ..., v_n)$ is the node set where v_i is the representation of sentence s_i. A node list of 20-sentences document is shown in Table 4.3. Thus, we create a comprehensive graph model that encompasses the entirety of a given document, its constituent sentences, and the degree of similarity between sentences. Figure 4.3 shows the graph presentation of the sample document consisting of 20 sentences.

In the next step, we move on to the stage of selecting the most important sentences(nodes) for inclusion in the summary.

Table 4.4: Edge weights based on SentenceBERT and cosine similarity for 20-sentence sample document obtained from the CNN/DM dataset.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
S1	0.000	0.189	-0.024	0.179	0.130	0.044	0.153	0.384	0.129	0.143	0.135	0.455	0.148	0.010	0.216	0.086	0.233	0.123	0.081	0.217
S2	0.189	0.000	0.422	0.177	0.073	0.061	0.098	0.274	0.289	0.362	0.332	0.357	0.277	0.384	0.221	0.110	0.292	0.076	0.251	0.062
S3	-0.024	0.422	0.000	0.143	0.161	0.136	0.025	0.087	0.215	0.111	0.338	0.062	-0.093	0.184	0.145	0.086	0.081	0.058	0.387	0.104
S4	0.179	0.177	0.143	0.000	0.216	0.515	0.248	0.396	0.080	0.136	0.046	0.166	-0.024	0.103	0.118	-0.061	0.018	0.009	0.102	0.099
S5	0.130	0.073	0.161	0.216	0.000	0.114	-0.064	0.057	0.153	0.162	0.046	-0.058	-0.058	0.083	0.071	0.031	-0.098	-0.002	0.019	0.149
S6	0.044	0.061	0.136	0.515	0.114	0.000	0.318	0.354	0.119	0.159	-0.063	0.050	-0.100	0.106	0.071	0.058	-0.002	-0.136	0.214	0.016
S7	0.153	0.098	0.025	0.248	-0.064	0.318	0.000	0.345	0.049	0.036	0.046	0.018	0.194	0.262	-0.070	0.045	0.276	-0.027	0.244	-0.001
S8	0.384	0.274	0.087	0.396	0.057	0.354	0.345	0.000	0.160	0.257	0.064	0.339	0.125	0.194	0.302	-0.039	0.213	0.009	0.091	0.009
S9	0.129	0.289	0.215	0.080	0.153	0.119	0.049	0.160	0.000	0.425	0.118	0.130	0.159	0.062	0.119	0.024	0.039	-0.003	0.039	-0.058
S10	0.143	0.362	0.111	0.136	0.162	0.159	0.036	0.257	0.425	0.000	0.181	0.203	0.210	0.008	0.244	0.020	0.158	0.029	0.019	0.040
S11	0.135	0.332	0.338	0.046	0.046	-0.063	0.046	0.064	0.118	0.181	0.000	0.282	0.259	0.154	0.154	0.246	0.165	0.215	0.257	0.108
S12	0.455	0.357	0.062	0.166	-0.058	0.050	0.018	0.339	0.130	0.203	0.282	0.000	0.372	0.122	0.407	0.170	0.452	0.197	0.112	0.239
S13	0.148	0.277	-0.093	-0.024	-0.058	-0.100	0.194	0.125	0.159	0.210	0.259	0.372	0.000	0.318	0.254	0.167	0.395	0.043	0.026	-0.033
S14	0.010	0.384	0.184	0.103	0.083	0.106	0.262	0.194	0.062	0.008	0.154	0.122	0.318	0.000	0.152	0.184	0.196	-0.040	0.348	-0.039
S15	0.216	0.221	0.145	0.118	0.071	0.071	-0.070	0.302	0.119	0.244	0.154	0.407	0.254	0.152	0.000	0.296	0.340	0.045	0.084	0.115
S16	0.086	0.110	0.086	-0.061	0.031	0.058	0.045	-0.039	0.024	0.020	0.246	0.170	0.167	0.184	0.296	0.000	0.190	0.149	0.205	0.326
S17	0.233	0.292	0.081	0.018	-0.098	-0.002	0.276	0.213	0.039	0.158	0.165	0.452	0.395	0.196	0.340	0.190	0.000	0.178	0.110	0.167
S18	0.123	0.076	0.058	0.009	-0.002	-0.136	-0.027	0.009	-0.003	0.029	0.215	0.197	0.043	-0.040	0.045	0.149	0.178	0.000	0.180	0.383
S19	0.081	0.251	0.387	0.102	0.019	0.214	0.244	0.091	0.039	0.019	0.257	0.112	0.026	0.348	0.084	0.205	0.110	0.180	0.000	0.254
S20	0.217	0.062	0.104	0.099	0.149	0.016	-0.001	0.009	-0.058	0.040	0.108	0.239	-0.033	-0.039	0.115	0.326	0.167	0.383	0.254	0.000



S1: Call it a little piece of heaven for a family torn apart by tragedy.
S2: Back in July, Sierra Sharry and Lane Smith were just about to become parents.
S3: Sharry was eight months pregnant.
S4: But then Smith fell and hit his head.
S5: He was taken to the OU Medical Center in Oklahoma City.
S6: Smith never recovered.
S7: "July 13th 2014 was the absolute worst day of my life," Sharry posted on Facebook.
S8: "I lost my best friend.
S9: The father of my unborn child."
S10: Their son Taos arrived a few weeks later.
S11: When it was time for his 6-month pictures, Sharry had a special request.
S12: Maybe the photographer could make their family complete, just for one picture .
S13: "They asked me if I would be willing to 'play around' with capturing their first family photo by editing Taos' daddy in one of their pictures, Kayli Rene' Photography posted on Facebook.'
S14: "I just got to thinking, we don't have a picture with Lane in it," the new mom told CNN affilaite KOCO.
S15: "one that has him looking over his family's shoulder."
S16: "Lane's not physically here with us, of course, but that picture represents to us that he is always watching over us and he will always be there for us no matter what," Sharry said.
S17: "The family photo has become a social media sensation after appearing on the photographer's Facebook page this week."
S18: It has some 193,000 likes and more than 24,000 shares.
S19: "I can't believe she actually did this," Sharry said.
S20: "It's like amazing and apparently everyone else thinks it is too."

Figure 4.3: The graph representation of 20-sentence sample document obtained from the CNN/DM dataset.

4.2.5 Ranking and Summary Selection

In graph-based summarisation methods, centrality is used to select the most salient sentence to construct summaries through ranking. Generally, undirected graph algorithms compute the sentence centrality score as follows:

$$Centrality(s_i) = \sum_{j=1}^{N} e_{ji} \tag{4.7}$$

After obtaining the centrality score for each sentence, sentences are sorted in reverse order and the top ranked sentences are included in the summary. GUSUM includes the node weights of the sentence graph in the calculation of the centrality. We propose a variation of weighted undirected graph-based ranking in this section. We modify Equation 4.7 to include the node weights. As a consequence, we define the importance rank for each sentence as follows:

$$Rank(s_i) = v[i] * \sum_{j=1}^{n} A[i, j] \tag{4.8}$$

We finally rank and select sentences with Equation 4.9. The number of sentences in the summary is represented by the k value.

$$summary = topK(\{Rank_{(si)})\}_{i=1,\ldots,n} \tag{4.9}$$

where the top-ranked k sentences will be extracted for the summary.

Table 4.5: Node weights after sentence centrality ranking for 20-sentence sample document obtained from the CNN/DM dataset.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
7.164	**11.427**	5.379	5.809	2.975	4.267	4.980	6.590	4.035	5.459	5.957	**7.008**	6.475	5.915	5.150	5.269	5.916	1.836	4.570	5.107

Table 4.5 presents the new node weights obtained after applying the centrality ranking of the document presented in Figure 4.3. According to the new rank values for creating a 3-sentence summary, Sentence 1, Sentence 2 and Sentence 12 are the sentences (nodes) that should be included in the summary.

4.3 Experimental Study

In this section we assess the performance of GUSUM on the document summarisation task. We first introduce the datasets that we used, then give our pre-processing and implementation details.

4.3.1 Summarisation Datasets

Table 4.6: Statistic of CNN/DM , NYT, PubMed and arXiv datasets.

Datasets	**#docs**	**#avg. doc. word**	**#avg. doc. sent.**	**#avg. sum. word**	**#avg. sum. sent.**
CNN/DM	11490	773.22	33.36	57.75	3.79
NYT	6508	1109.10	32.17	96.31	1.18
PubMed	6658	3142.92	101.60	208.02	7.58
arXiv	6440	6446.10	250.36	166.72	6.22

CNN/DM dataset contains 93k articles from CNN, and 220k articles from Daily Mail newspapers, which uses their associated highlights as reference summaries (Hermann et al., 2015). We use the test set which includes 11490 documents provided by hugging face version 3.0.0[2] (See et al., 2017).

NYT dataset contains over 1.8 million articles written and published by the New York Times between January 1, 1987 and June 19, 2007 and summaries are written by library scientists. Different from CNN/DM, salient sentences are distributed evenly in each article. We use The New York Times Annotated Corpus provided by the Linguistic Data Consortium[3] (Sandhaus, 2008). We filter out documents whose summaries are between January 1, 2007 and June 19, 2007 and documents whose length of summaries are shorter than 50 tokens and finally retain 6508 documents (Zheng & Lapata, 2019) .

[2] https://huggingface.co/datasets/cnn_dailymail

[3] https://catalog.ldc.upenn.edu/LDC2008T19

PubMed & arXiv datasets are two long documents datasets of scientific papers. The datasets are obtained from arXiv and PubMed OpenAccess repositories. The summaries are created from the documents. PubMed contains 215k and arXiv contains 113k documents. We use test set which includes 6658 documents for PubMed and 6440 documents for arXiv provided by hugging face[4].

4.3.2 Implementation Details

In GUSUM, during the pre-processing stage, NLTK (Natural Language Toolkit)(Bird & Loper, 2004) were used to collect corpus statistics and process documents using methods such as sentence segmentation, word tokenisation, Part of Speech (POS) tagging and using regular expressions to remove parenthesis and some characters. In the process of creating the graph, we first apply Equations 4.1, 4.2, 4.3 and 4.4 to calculate sentence feature scores and define the sums of the obtained values as node weights. Next, we calculate the edge weights representing the sentence similarities. For each dataset, we use the publicly released SentenceBERT model *roberta-base-nli-stsb-mean-tokens* [5] to initialise our sentence embeddings. The *bert-base-nli-mean-tokens*[6] model was also tested in our experiments. However, the *roberta-base-nli-stsb-mean-tokens* showed slightly higher performance (See Table 4.12). Alternative models that can be applied in our method are listed on Github[7].

In our experiments, Cosine distance and Euclidean distance are tested to measure the distances between sentence embedding vectors. However, it is observed that higher performance is obtained with the Cosine similarity (Equation 4.5) method when using SentenceBERT (Table 4.12). The scores obtained from the similarity measure are assigned to be the edge weights of the graph. In the last stage, we rank the sentences using Equation 4.7

[4] https://www.tensorflow.org/datasets/catalog/scientific_papers

[5] https://huggingface.co/sentence-transformers/roberta-base-nli-stsb-mean-tokens

[6] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

[7] https://github.com/tubagokhan/GUSUM/blob/main/QAforHumanEvaluation.json

and determine the three most important sentences that should be included in the summary. Table 4.7 presents a sample gold standard reference summary and the summary created by GUSUM.

Table 4.7: Comparison of an example document from CNN/Daily Dataset (News 615) and the summary generated by GUSUM.

Original Document (Bold sentences are the sentences included in the summary by GUSUM).	
S1	**Fewer tourists and relatively warm temperatures may be reason enough to put Ireland on your list of winter travel destinations, especially Dingle Peninsula, once ranked by National Geographic Traveler as "the most beautiful place on Earth."**
S2	**Winter offers tourists a chance to explore Ireland's west coast unhindered by bothersome crowds.**
S3	The peninsula, on Ireland's west coast, includes the oceanside town of Dingle, which boasts more than 1,000 full-time residents.
S4	Winter visitors will avoid the area's hundreds of thousands of summertime tourists.
S5	Boats crowd Dingle's popular marina, bringing fresh seafood catches of the day.
S6	Some of the marina vessels also will ferry visitors to see Fungie, locally famous dolphin who has lived in the waters outside town since 1984.
S7	See breathtaking photos of Dingle.
S8	Outside Dingle, numerous vacation cottages are available to rent, including homes in the village of Dunquin.
S9	In winter, rates are drastically cut, and rental period dates may be more flexible.
S10	Most shops and restaurants have shorter hours during winter, and traditional music is found in some of the pubs on the weekends.
S11	As with most of Ireland, pubs abound, even in the smallest villages.
S12	A beer and some hearty pub grub are a perfect way to cap a day of exploring the wintry sights of the peninsula.
S13	**Because Ireland sits near the warm waters of the Atlantic Gulf Stream, the Emerald Isle has an average temperature of 46 degrees Fahrenheit during December, January and February.**
S14	**But pack smart and bring layers of clothing, including warm sweaters and jackets, because winter weather often means rain on Ireland's western shore.**
Gold-Standard Reference (Sentences with ++ are sentences that match the summary created by GUSUM.)	
S1	Dingle, Ireland, called "most beautiful place on Earth" by National Geographic.{++}
S2	Escape summer crowds by traveling to Dingle Peninsula during winter months. {++}
S9	Cottage rentals are cheaper in winter, and periods are more flexible.
S13	Winter temperatures in western Ireland average 46 degrees F (7 Celsius). {++}

As seen in Table 4.7, the example generates a summary consisting of 4 sentences. When we manually evaluated the document, we can see the Sentences 1, 2, 9, and 13 should be captured by an extractive summarisation system based on the gold standard reference summary. Upon evaluation, GUSUM successfully identifies sentences 1, 2, and 13, but ,on the other side, includes sentence 14 instead of sentence 9.

4.4 Results

4.4.1 Automated Evaluation

Table 4.8 and Table 4.9 summarise our results for the CNN/DM and NYT short document datasets and arXiv and PubMed long document datasets respectively. The first blocks present the results from the unsupervised baselines LEAD-3, TEXTRANK (Mihalcea & Tarau, 2004)), LEXRANK (Erkan & Radev, 2004) previous unsupervised graph-based methods. LEAD simply selects the first sentences to be the summary for each document. TEXTRANK displays a document as a graph with sentences as nodes, edge weights are calculated using sentence similarity and a modification of the PageRank algorithm (Brin & Page, 1998) is used to select the best scores. LEXRANK also calculates the significance of sentences in representative graphs based on a measure of eigenvector centrality. The second block shows recently published supervised methods. For supervised extractive models, we compare with EXTRACTION (J. Xu & Durrett, 2019), REFRESH (Narayan et al., 2018a), BertEx (Y. Liu & Lapata, 2019), Discourse-aware (Cohan et al., 2018), SummaRuNNer (Nallapati et al., 2017) and GlobalLocalCont (Xiao & Carenini, 2019). The third blocks includes recent state-of-the-art unsupervised graph-based methods for document summarisation. PACSUM (Zheng & Lapata, 2019), FAR (Liang et al., 2021), STAS (S. Xu et al., 2020) and J. Liu et al. (2021) were detailed in Chapter 2.3.2. The last blocks in Table 4.8 and Table 4.9 report results for our method, GUSUM.

Table 4.8 shows that GUSUM achieves the highest ROUGE F1 score, compared to all other graph-based unsupervised methods on both the CNN/DM and NYT datasets. From the results, we can see that our method outperforms all strong baselines in the first block. Furthermore, our method achieves better results than PACSUM and FAR on both datasets. When we compare our method with STAS, our method produces better results, except for the F-1 R-2 metric on CNN/DM. The success of GUSUM can be seen when the latest state-of-the-art unsupervised graph-based method by J. Liu et al. (2021) and GUSUM are compared. Moreover, it is seen in Table 4.9, GUSUM also performed very well on arXiv and PubMed long document datasets especially for F1 R-L.

Table 4.8: ROUGE/F1 scores of GUSUM on the CNN/DM and NYT datasets.

		CNN/DM			**NYT**		
Category	Method	R-1	R-2	R-L	R-1	R-2	R-L
Baselines	LEAD-3	40.49	17.66	36.75	35.50	17.20	32.00
	TEXTRANK (Mihalcea & Tarau, 2004)	33.85	13.61	30.14	33.24	14.74	29.92
	LEXRANK (Erkan & Radev, 2004)	34.68	12.82	31.12	30.75	10.49	26.58
Supervised	EXTRACTION (J. Xu & Durrett, 2019)	40.70	18.00	36.80	44.30	25.50	37.10
	REFRESH (Narayan et al., 2018a)	41.30	18.40	35.70	41.30	22.00	37.80
	BertExt (Y. Liu & Lapata, 2019)	43.25	20.24	39.63	-	-	-
Unsupervised	PACSUM (Zheng & Lapata, 2019)	40.70	17.80	36.90	41.40	21.70	37.50
	FAR (Liang et al., 2021)	40.83	17.85	36.91	41.61	21.88	37.59
	STAS (S. Xu et al., 2020)	40.90	18.02	37.21	41.46	21.80	37.57
	Liu (J. Liu et al., 2021)	41.60	**18.50**	37.80	42.20	21.80	38.20
Current Work	GUSUM	**43.40**	17.02	**42.38**	**43.64**	**22.01**	**37.90**

* Results are taken from (Liang et al., 2021). Underlined values indicate the highest values in supervised methods, and bold values indicate the highest values in unsupervised methods.

Table 4.9: ROUGE/F1 scores of GUSUM on the arXiv and PubMed datasets.

		arXiv			**PubMed**		
Category	Method	R-1	R-2	R-L	R-1	R-2	R-L
Baselines	LEAD	33.66	8.94	22.19	35.63	12.28	25.17
	TEXTRANK (Mihalcea & Tarau, 2004)	24.38	10.57	22.18	38.66	15.87	34.53
	LEXRANK (Erkan & Radev, 2004)	33.85	10.73	28.99	39.19	13.89	34.59
Supervised	Discourse-aware (Cohan et al., 2018)	35.80	11.05	31.80	38.93	15.37	35.21
	SummaRuNNer (Nallapati et al., 2017)	42.81	16.52	28.23	43.89	18.78	30.36
	GlobalLocalCont (Xiao & Carenini, 2019)	43.62	17.36	29.14	44.85	19.70	31.43
Unsupervised	PACSUM (Zheng & Lapata, 2019)	39.33	12.19	34.18	39.79	14.00	36.09
	FAR (Liang et al., 2021)	40.92	**13.75**	35.56	41.98	15.66	37.58
Current Work	GUSUM	**40.98**	11.76	**39.49**	**44.98**	**16.26**	**43.98**

* Results are taken from (Liang et al., 2021). Underlined values indicate the highest values in supervised methods, and bold values indicate the highest values in unsupervised methods.

4.4.2 Human Evaluation

In addition to the Rouge metric, we also evaluated the system output via human judgements. In the experiment, we evaluated the extent to which our approach retained important information in the document, following a question-answer (QA) paradigm used to evaluate the summary quality and text compression (Narayan et al., 2018a).

Table 4.10: A sample news, gold summary from NYT dataset and human evaluation questions and answers for sample document.

Document:

Tom Oreck, the chief executive of the vacuum cleaner maker founded by his father, David, has a whole lot of new floor space to keep clean. The younger Mr. Oreck paid $4.6 million this month for a 7,800-square-foot mansion in Nashville that was once owned by the Grand Ole Opry star Minnie Pearl and her husband, Henry Cannon. The home sits on 4.7 acres near the Tennessee governor's mansion.Mr. Oreck's purchase rekindled speculation that the company might move its headquarters from New Orleans to Tennessee, where it relocated some of its operations after Hurricane Katrina.Mr. Oreck said in an interview that although he and his family would move to Nashville, the ”official headquarters of the company is still in New Orleans,” where his father and his uncle, Marshall, have their offices.PATRICK McGEEHANOPENERS: SUITS

Gold Summary:

Tom Oreck, chief executive of vacuum cleaner company, has bought mansion in Nashville that was once owned by late Grand Ole Opry star Minnie Pearl, rekindling speculation that company might move its headquarters from New Orleans to Tennessee; Oreck says official headquarters of company is still in New Orleans although he and his family will move to Nashville; photo (S)

Questions:

1) Who is the chief executive of vacuum cleaner company?
2) Where is his new mansion?
3) Who is the old owner of the mansion?

Answers:

1) Tom Oreck
2) Nashville
3) Minnie Pearl

We created a set of questions based on the assumption that gold-standard summaries. Then, we examined whether participants could answer these questions simply by reading the system summaries without accessing the article. We created 71 questions from 20 randomly selected documents for the CNN/DM datasets and 59 questions from 18 randomly selected documents for the NYT dataset. We wrote multiple fact-based question-answer pairs for each gold summary. A sample question and answer set shown in 4.10. Our all question and answer set is available at https://github.com/tubagokhan/GUSUM/blob/main/QAforHumanEvaluation.json.

We compared GUSUM against LEAD-3 and TEXTRANK on CNN/DM and NYT. We used the same scoring mechanism from Ziheng and Lapata (2019), a correct answer was marked with a score of one, partially correct answers with a score of 0.5, and zero otherwise. The final score for a system is the average of all its question scores. Four fluent English speakers answered the questions for each summary. The participants were chosen from university volunteers who gave their consent to contribute to the study.

Table 4.11: QA-based human evaluation results of GUSUM on CNN/DM and NYT.

Method	**CNN/DM**		**NYT**	
	Score	%	Score	%
LEAD-3	54.75	77.11	42.00	71.19
TEXTRANK	56.38	79.40	39.50	66.95
GUSUM	**57.00**	**80.28**	**46.25**	**78.39**

* We compute a system's final score as the average of all question scores.

The results of our QA evaluation are shown in Table 4.11. Based on summaries generated by LEAD-3 participants answered 77.11% and 71.19% correctly answered questions from CNN/DM and NYT correctly. Summaries produced by TEXTRANK have 79.40% and 66.95% scores. When the scores of GUSUM are compared with the scores of the other two systems, the high performance of GUSUM is seen (80.28% and 78.39%). The main reason for GUSUM's slightly higher performance in CNN/DM dataset compared to NYT is thought to be the use of human-generated gold summaries in NYT. Another possibility is that the summaries created from the CNN/DM dataset are shorter and users can focus more. It is thought that the participants had a tendency to become distracted with the longer summaries in the NYT dataset compared to CNN/DM.

4.4.3 Ablation Study

In order to access the contribution of three components of GUSUM, we remove or change each component of them one at a time and report ablation study results in Table 4.12. In Table 4.12, the results of the NYT dataset in the first block and the PubMed dataset in the second block are presented.

Table 4.12: ROUGE/F1 results of GUSUM ablation study on NYT and PubMed datasets.

NYT			
	R-1	R-2	R-L
GUSUM	**43.64**	**22.01**	**37.90**
-Removed All Sentence Features	36.63	14.91	30.58
-bert-base-nli-mean-tokens	43.28	21.73	37.48
-Eucludian Distance	35.35	16.43	31.10
PubMed			
GUSUM	**44.98**	**16.26**	**43.98**
-Removed All Sentence Features	44.08	15.53	43.32
-bert-base-nli-mean-tokens	44.27	15.66	43.36
-Eucludian Distance	37.77	11.29	37.40

The results suggest that that sentence feature scoring is critical to GUSUM's performance in short document summarisation for NYT. When all sentence features are eliminated (assigned 1), the performance of GUSUM drops sharply. In contrast, when we replaced the *roberta-base-nli-stsb-mean-tokens* model with the *bert-base-nli-mean-tokens* model in both datasets there was little difference in the results. In our last experiment, we changed the method of measuring the similarity of sentence embeddings to generate the graph. When we employ the Euclidean method, there is a dramatic decrease in the performance of GUSUM in both datasets.

4.5 Discussion

This research aims to demonstrate the importance of node weights in graph-based text summarisation systems. Our proposed method shows the effect of embedding node weights in graph models used for text summarisation on the performance of the summarisation systems. As seen in the results, GUSUM showed high performance on all datasets. Inclusion of sentence feature scores led to significant improvement in the scores from rouge metrics for short documents. However, for long documents the inclusion of sentence features only marginally improved rouge scores in the test datasets. However, the ranking algorithm applied in this chapter was fairly simple and there may be potential for further improvements in long document performance by considering how the node weights are incorporated into the ranking system

The most challenging part of this study is the evaluation stage. Evaluating the performance of summarisation systems poses a problem for many researchers (Schluter, 2017). The results and limits of the commonly used methods for automatic evaluation methods are a matter of debate. In addition, it is a known fact by researchers that human evaluation is the best summary performance evaluation method. For this reason, we included human evaluation as a performance evaluation method in our study. However, we noticed in our study that the questions used for human evaluation based on the QA paradigm in other studies published to date have yet to be shared by the researchers. As a result of this situation, researchers prepare their questions in human-based evaluations, and the results cannot be compared with the literature. As a solution to this problem, we publish the questions and answers we prepared from the CNN/DM and NYT datasets based on the QA paradigm for future studies (See 4.4.2) and we recommend that in the future other researchers will do the same.

4.6 Conclusion

In this section, we have introduced a node-weighted graph model. We propose a graph-based single-document unsupervised extractive summarisation method to demonstrate the effect of node weights in graph-based summarisation systems. We define values indicating the importance of the sentences in the document for the node weights in the graphs. We also build graphs with undirected edges by employing SentenceBERT to capture sentence similarity better. Experimental results on four summarisation benchmark datasets demonstrated that our method outperforms other recently proposed extractive graph-based unsupervised methods and produces comparable results with other supervised methods. The next chapter describes our novel ranking algorithm proposed to overcome the limitations of GUSUM in the long document summarisation.

Chapter 5

Node-Weighted Centrality Ranking for Unsupervised Long Document Summarisation

In this chapter, we introduce a graph-based unsupervised extractive long document summarisation approach based on a novel node-weighted graph ranking algorithm. Section 5.1 formulates the problem, and we further discuss the details of the methodology in Section 5.2. Section 5.3 presents the dataset and the experimental setup. Finally, the results are presented in Section 5.4, followed by some discussion and the conclusion of the chapter in Sections 5.5 and 5.6, respectively.

The work described in this chapter was published in the paper Node-Weighted Centrality Ranking for Unsupervised Long Document Summarisation in Natural Language Processing and Information Systems: 28th International Conference on Applications of Natural Language to Information Systems (NLDB 2023) (Gokhan et al., 2023).

5.1 Introduction

Unsupervised graph-based methods entail a node selection task for summarisation, in which nodes or sentences are chosen for the final summary after creating graphs. The centrality concept is employed to determine the importance of a node in the graph. TEXTRANK (Mihalcea & Tarau, 2004) and LEXRANK (Erkan & Radev, 2004) serve as prominent examples of early work in extractive graph-based summarisation. They represent sentences as nodes in an undirected graph, with edge weights based on sentence occurrence similarity. Graph-based ranking algorithms such as PageRank are used to determine the final sentence ranking scores.

Researchers have continued to explore the potential of unsupervised graph-based methods in summarisation after the early success of graph-based summarisation approaches such as TEXTRANK and LEXRANK. Zheng and Lapata (2019)'s PACSUM is one of the leading works of recent years, revisiting sentence centrality for unsupervised summarisation. Zheng and Lapata create a directed graph using BERT to compute sentence similarities. The sentence centrality score of a sentence is the weighted sum of all its outer edges, where weights for edges between the current sentence and preceding sentences are negative. The edges in the directed graph represent the relative position of the sentences in the document. In our study, similar to Zheng and Lapata, the edges represent sentence similarities, but we evaluate the nodes from a different perspective by incorporating sentence features into the nodes. Although Zheng and Lapata, as well as other methods discussed in Section 2.3.2, use PageRank (Brin & Page, 1998) as a centrality ranking algorithm, this approach may not provide the level of precision required for a comprehensive analysis of fully-weighted graphs. Especially for graph models representing long documents, which have larger and more complex structures compared to graphs representing short documents. Therefore, a more advanced sentence centrality ranking method may be required.

In this study, to improve long document extractive summarisation performance, we propose a novel **No**de-**W**eighted Centrality Ranking Approach for Unsupervised Graph-Based Long Document Extractive **Sum**marisation (**NoWRANK**). In our approach, we create an undirected fully weighted graph model for each document. First, to define augmented node weights (i.e., sentences). For augmentation, we use two well-known summarisation methods for our two different pipelines: Latent Semantic Analysis (Dumais, 2004) and Sentence Feature Scoring (Suanmali, Salim, & Binwahlan, 2009). Secondly, we employ SentenceBERT (Reimers & Gurevych, 2019) to better capture sentence meaning and compute sentence similarity and define the weight of edges. Finally, we apply our novel Node-Weighted Centrality Ranking method to the node-weighted graphs. NoWRANK is developed based on eigenvector centrality (Ruhnau, 2000) by including node-weights. We evaluate our approach to the summarisation of long scientific documents from PubMed and arXiv (Cohan et al., 2018). Our experimental results demonstrate that our method outperforms earlier state-of-the-art unsupervised graph-based summarisation algorithms and surpasses strong unsupervised baselines. In addition, our straightforward, unsupervised method also shows performance equivalent to that of state-of-the-art supervised neural models trained on large documents.

5.2 Methodology

In this section, we present our methodology for long document extractive summarisation by introducing a ranking algorithm and the calculation of node and edge weights for the proposed graph model. (See Figure 5.1)

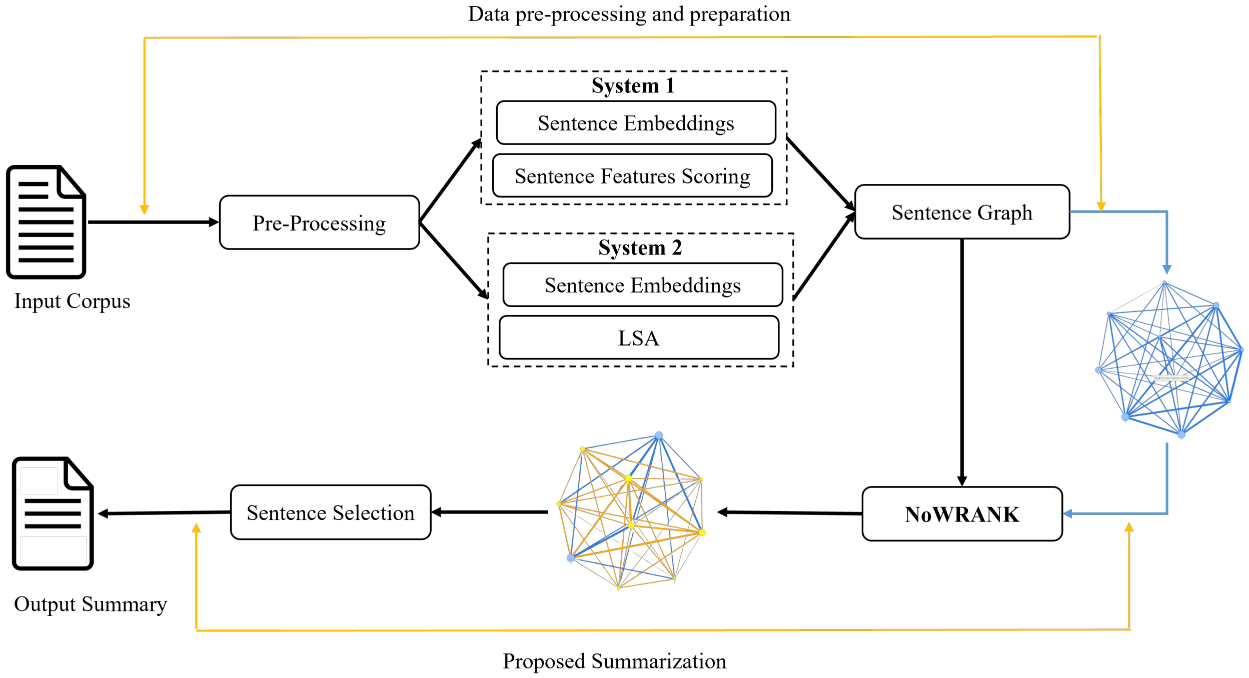


Figure 5.1: Our graph-based summarisation system pipeline using our node-weighted ranking algorithm.

5.2.1 Calculation of Node Weights

The proposed node-weighted graph model is developed to easily integrate with different approaches. The critical point is to define the characteristic features of the sentences in the document, independent of similarity, and in the most distinctive way by means of node weights. These characteristic features can be measured using different methods. Our system evaluates two statistical text summarisation approaches with a new perspective and assigns node weights.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) examines the semantic similarity between different texts by using a statistical model of word usage (Landauer et al., 1998). The concept of employing

LSA for text summarisation was first published by Gong and Liu (2001) and they use singular value decomposition to summarise generic text. Since then, various LSA-based summarising techniques have been developed Babar and Thorat, 2014; Cagliero et al., 2019; Gupta and Patel, 2021; John et al., 2017; Ozsoy et al., 2010; Steinberger and Jezek, 2004.

This study analyses a corpus using the LSA algorithm to identify node weights. The corpus is first parsed into individual sentences, and a term-sentence matrix is created that represents the frequency of each word in each sentence. The term-sentence matrix is decomposed using singular value decomposition (SVD).

SVD is a mathematical technique used in LSA to reduce the dimensionality of the term-sentence matrix and identify the underlying latent semantic structure. SVD is used to factorise this matrix into three matrices - U: a left singular matrix, S: a diagonal singular value matrix, and V: a right singular matrix. In LSA, SVD takes a rectangular term-sentence matrix of n data (defined as A, where A is an nxp matrix) in which the n rows represent the terms, and the p columns represent the sentences. The SVD theorem states:

$$A_{nxp} = U_{nxn} S_{nxp} V^T_{pxp} \tag{5.1}$$

The left singular matrix contains information about the relationship between the rows of the original matrix. The diagonal singular value matrix contains information about the importance of each dimension in the original matrix. The right singular matrix contains information about the relationship between the columns of the original matrix (Figure 5.2). The SVD contains the importance of each dimension or latent semantic space, with the highest values representing the most important dimensions. Thus, SVD can reduce the dimensionality of the original matrix while retaining most of the information. This process is called dimensionality reduction, and it helps to remove noise and extract the most important features or latent semantic information from the original matrix.

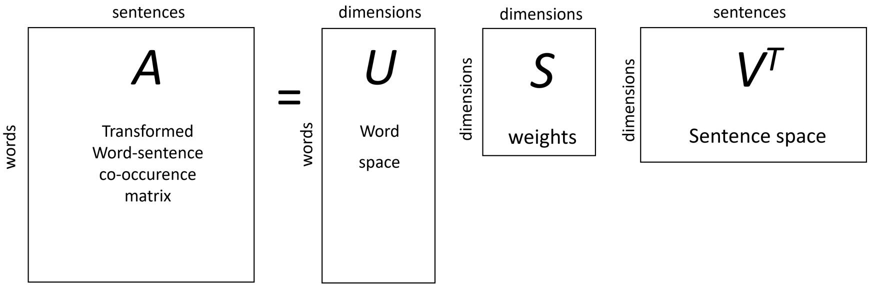


Figure 5.2: Singular Value Decomposition of a matrix.

In our first approach, each sentence in the corpus is subsequently scored based on the reduced matrix. Sentences with the highest similarity to these concepts are regarded as the most important and are allocated higher scores. The sentences in the corpus are then ranked according to their scores, with the highest-scoring sentences being considered the most important. The values obtained from this process are then used to assign node weights. Table 5.1 shows the node weights for a sample document.

Sentence Features Scoring

In our second approach, we use sentence feature scoring methods to define the node weights. In the sentence feature scoring method, features of sentences are analysed and then combined together to obtain the score for each sentence. Sentence scoring methods can involve the presence of many features including term frequency, noun and verb phrases, content word, title word, proper noun, cue-phrase, numerical data, sentence location, sentence length. We focused on four features for each sentence, these are *Sentence length*, *Sentence position*, *Proper nouns* and *Numerical token*. The sum of these calculated values was defined as the relevant node weight (Section 4.2.2). Table 5.1 shows the node weights for a sample long document obtained from arXiv dataset.

Table 5.1: A sample document from the arXiv dataset and the sentences' node weights.

Corpus	LSA	Sentence Feature
for about 20 years the problem of properties of short - term changes of solar activity has been considered	10.2684	2.1624
extensively . many investigators studied the short - term periodicities of the various indices of solar activity .	10.3238	1.1559
several periodicities were detected , but the periodicities about 155 days and from the interval of @xmath3	10.4278	2.3075
]$days(@xmath4$] years) are mentioned most often . first of them was discovered by @xcite in the occurence rate	10.3615	1.3945
of gamma - ray flares detected by the gamma - ray spectrometer aboard the _ solar maximum mission (smm) .	10.3470	1.2747
this periodicity was confirmed for other solar flares data and for the same time period @xcite . it was also found in	10.3769	2.3398
proton flares during solar cycles 19 and 20 @xcite , but it was not found in the solar flares data during solar cycles	10.5281	1.1054
22 @xcite . _ several autors confirmed above results for the daily sunspot area data . @xcite studied the sunspot	10.3267	2.0613
data from 18741984 . she found the 155-day periodicity in data records from 31 years . this periodicity is always	10.4480	2.0039
characteristic for one of the solar hemispheres (the southern hemisphere for cycles 1215 and the northern	10.5281	1.9998
hemisphere for cycles 1621) . moreover , it is only present during epochs of maximum activity (in episodes of 13	10.4881	1.0790
years) . similarinvestigationswerecarriedoutby + @xcite . they applied the same power spectrum method as lean ,	10.1627	1.6049
but the daily sunspot area data (cycles 1221) were divided into 10 shorter time series . the periodicities were	10.4264	2.1492
searched for the frequency interval 57115 nhz (100200 days) and for each of 10 time series . the authors showed	10.3397	2.0156
that the periodicity between 150160 days is statistically significant during all cycles from 16 to 21 . the considered	10.3470	1.9926
peaks were remained unaltered after removing the 11-year cycle and applying the power spectrum analysis .	10.2489	1.1185
@xcite used the wavelet technique for the daily sunspot areas between 1874 and 1993 . they determined the	10.2830	1.9694
epochs of appearance of this periodicity and concluded that it presents around the maximum activity period in	10.4508	2.1003
cycles 16 to 21 . moreover , the power of this periodicity started growing at cycle 19 , decreased in cycles 20 and	10.3875	2.0162
21 and disappered after cycle 21 . similaranalyseswerepresentedby + @xcite , but for sunspot number , solar wind	10.7257	1.3288
plasma , interplanetary magnetic field and geomagnetic activity index @xmath5 . during 1964 - 2000 the sunspot	10.3712	2.2939
number wavelet power of periods less than one year shows a cyclic evolution with the phase of the solar cycle.the	10.2830	1.1973
154-day period is prominent and its strenth is stronger around the 1982 - 1984 interval in almost all solar wind	10.2347	1.0298
parameters . the existence of the 156-day periodicity in sunspot data were confirmed by @xcite . they considered	10.3209	1.2498
the possible relation between the 475-day (1.3-year) and 156-day periodicities . the 475-day (1.3-year)	10.3875	1.1751
periodicity was also detected in variations of the interplanetary magnetic field , geomagnetic activity helioseismic	10.2757	2.1331
data and in the solar wind speed @xcite . @xcite concluded that the region of larger wavelet power shifts from	10.3078	1.1159
475-day (1.3-year) period to 620-day (1.7-year) period and then back to 475-day (1.3-year) . the periodicities	10.5281	2.0293
from the interval @xmath6]$days(@xmath4$] years) have been considered from 1968 . @xcite mentioned a	10.5281	1.0152
16.3-month (490-day) periodicity in the sunspot numbers and in the geomagnetic data . @xcite analysed the	10.4594	2.1711
occurrence rate of major flares during solar cycles 19 . they found a 18-month (540-day) periodicity in flare rate	10.5679	1.9882
of the norhern hemisphere . @xcite confirmed this result for the @xmath7 flare data for solar cycles 20 and 21 and	10.4523	1.0930
found a peak in the power spectra near 510540 days . @xcite found a 17-month (510-day) periodicity of sunspot	10.3629	2.1589
groups and their areas from 1969 to 1986 . these authors concluded that the length of this period is variable and	10.3962	2.0148
the reason of this periodicity is still not understood . @xcite and + @xcite obtained statistically significant peaks	10.2928	1.1807
of power at around 158 days for daily sunspot data from 1923 - 1933 (cycle 16) . in this paper the problem of the	10.2669	1.9096
existence of this periodicity for sunspot data from cycle 16 is considered . the daily sunspot areas , the mean	10.3716	1.9898
sunspot areas per carrington rotation , the monthly sunspot numbers and their fluctuations , which are obtained	10.3015	2.0467
after removing the 11-year cycle are analysed	...	...

As shown in Table 5.1, there is a considerable difference between the node weights obtained from the two systems. The values obtained with LSA are within a more limited range than sentence feature scoring method. For example, in this document, the node weights vary around the value of 10. On the other hand, the node weights obtained with sentence feature scoring always form a value between 0 and 4. The observed difference between the node weights obtained from the two systems serves as a basis for evaluating the performance of our ranking method, as it enables a more nuanced assessment of the impact of node weights.

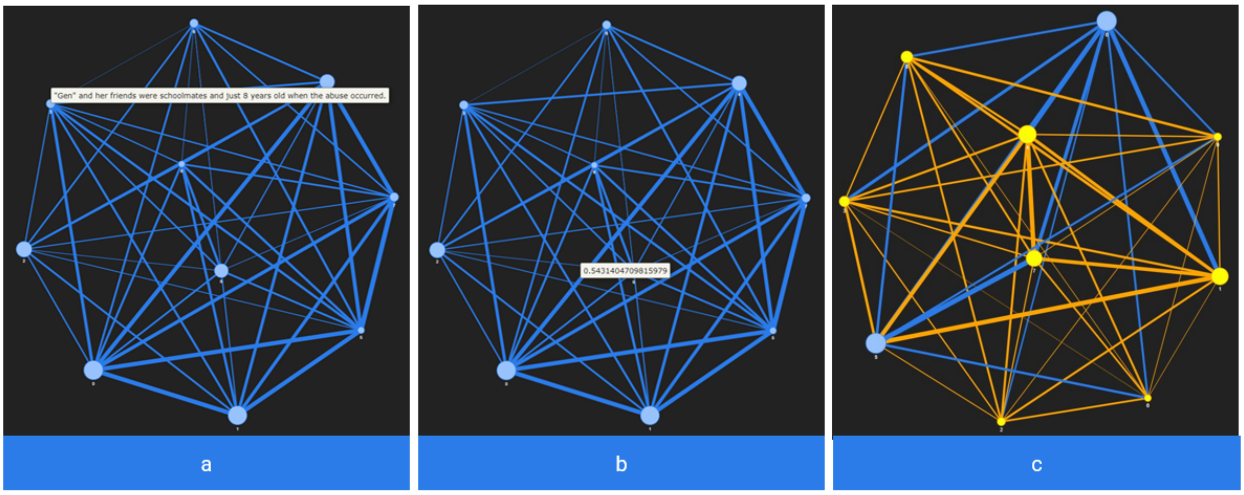

"Gen" and her friends were schoolmates and just 8 years old when the abuse occurred.
0.5431404709815979
a
b
c

Figure 5.3: Displaying a text consisting of 10-sentences on a node-weighted graph and selecting a 2-sentences summary via NoWRANK (a) Assigning node weights, (b) Assigning edge weights, (c) Displaying the most important nodes selected by NoWRANK. The sample is selected from the CNN/DM dataset.

5.2.2 Calculation of Edge Weights

In this step, we follow the previous studies to create our graph model and define the edge weights to capture similarity between sentences. An example graph model is also visualised in Figure 5.3. The graph visualized with pyvis [1]library. For creating edge weights, we use the publicly released SentenceBERT model *roberta-base-nli-stsb-mean-tokens* [2] for each dataset to initialise our sentence embeddings. The model maps sentences to a high dimensional

[1] https://pyvis.readthedocs.io/en/latest/index.html

[2] https://huggingface.co/sentence-transformers/roberta-base-nli-stsb-mean-tokens

dense vector space. Sentence similarities are calculated by applying the Cosine Distance measurement between the sentence embedding vectors. Figure 5.3(B) illustrates the graph generated through assigning the similarity scores between the sentence pairs to the edge weights.

5.2.3 Ranking via NowRANK

Given a document D, which contains a set of sentences $(s_1, s_2, ..., s_n)$and can be represented D as a graph $G = (V; E)$. $V = (v_{s1}, v_{s2}..., v_{sn})$ is the node (vertex) set where v_{si} is the representation of sentence s_i. $E = e[s_i; s_j]$ is the matrix set where $e[s_i; s_j]$ the representation of relationship between sentences s_i and s_j.

Eigenvector centrality is calculated on the basis of the adjacency matrix. The adjacency matrix is formed by Equation 5.2.

$$A_{i,j} = \begin{cases} 1 & \text{if node i is linked to node j} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

Eigenvector centrality for $vertex_i$, c_i, can be defined as:

$$c_i = \frac{1}{\lambda} \sum_{j \epsilon M(i)}^{N} c_j = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} c_j \tag{5.3}$$

where $M(v)$ is the set of neighbours of i and λ is a constant. If we reorder Equation 5.3 and express it in matrix form, we obtain an eigenvalue equation:

$$\mathbf{Ac} = \lambda \mathbf{c} \tag{5.4}$$

with λ and c being the eigenvalue and eigenvector, respectively.

In order to provide a clearer understanding of the calculation process of eigenvector centrality, we present a sample calculation in this section. This example aims to illustrate the application of eigenvector centrality in a practical context, and to demonstrate how the algorithm computes the relative importance of nodes based on the connectivity patterns of the graph. To illustrate the eigenvector centrality calculation, we consider a simple undirected graph with five nodes, as depicted in Figure 5.4.

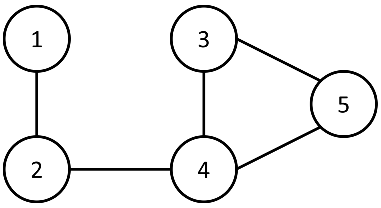


Figure 5.4: A simple undirected graph with five nodes

$$AdjacencyMatrix = A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\text{Iteration 1: } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} x \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 0.213 \\ 0.426 \\ 0.426 \\ 0.639 \\ 0.426 \end{bmatrix}$$

$$NormalisationValue = \sqrt{1^2 + 2^2 + 2^2 + 3^2 + 2^2} = 4.69$$

$$\text{Iteration 2: } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.213 \\ 0.426 \\ 0.426 \\ 0.639 \\ 0.426 \end{bmatrix} = \begin{bmatrix} 0.426 \\ 0.852 \\ 1.065 \\ 1.278 \\ 1.065 \end{bmatrix} \equiv \begin{bmatrix} 0.195 \\ 0.389 \\ 0.486 \\ 0.584 \\ 0.486 \end{bmatrix}$$

$$NormalisationValue = 2.19$$

$$\text{Iteration 3: } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.195 \\ 0.389 \\ 0.486 \\ 0.584 \\ 0.486 \end{bmatrix} = \begin{bmatrix} 0.389 \\ 0.779 \\ 1.07 \\ 1.361 \\ 1.07 \end{bmatrix} \equiv \begin{bmatrix} 0.176 \\ 0.352 \\ 0.484 \\ 0.616 \\ 0.484 \end{bmatrix}$$

$$NormalisationValue = 2.21$$

$$\text{Iteration 4: } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.176 \\ 0.352 \\ 0.484 \\ 0.616 \\ 0.484 \end{bmatrix} = \begin{bmatrix} 0.352 \\ 0.792 \\ 1.100 \\ 1.320 \\ 1.100 \end{bmatrix} \equiv \begin{bmatrix} 0.176 \\ 0.352 \\ 0.484 \\ 0.616 \\ 0.484 \end{bmatrix}$$

$NormalisationValue = 2.21$ converges

After the fourth iteration, the normalisation values converged, indicating the completion of the eigenvector centrality calculation process. The eigenvector centrality vector is $[0.176, 0.352,\ 0.484, 0.616, 0.484]$. Thus, node ranking scores are $1, 2, 3, 5, 4$ with node 4 determined to be the most important node on the graph.

In our method, we include the node weights during the creation of the adjacency matrix. We reformulate the associated adjacency matrix as:

$$A_{[i,j]} = e_{[si,sj]} + (v_{si} + v_{sj}) * k \tag{5.5}$$

where A_{ij} is a symmetric NxN matrix, N is the total number of nodes, and k is constant value for normalisation. The values of an example graph, and the adjacency matrix created by our method for this chart are shown below. The k value is assigned to be 0.01.

$$V = [2.16, 1.15, 2.30, 1.39, 1.270] \qquad E = \begin{bmatrix} 0 & 0.18 & 0.02 & 0.17 & 0.13 \\ 0.18 & 0 & 0.42 & 0.17 & 0.07 \\ 0.02 & 0.42 & 0 & 0.14 & 0.16 \\ 0.17 & 0.17 & 0.14 & 0 & 0.21 \\ 0.13 & 0.07 & 0.16 & 0.21 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.0432 & 0.2131 & 0.0646 & 0.2055 & 0.1643 \\ 0.2131 & 0.0230 & 0.4545 & 0.1954 & 0.0942 \\ 0.0646 & 0.4545 & 0.0460 & 0.1769 & 0.1957 \\ 0.2055 & 0.1954 & 0.1769 & 0.0278 & 0.2366 \\ 0.1643 & 0.0942 & 0.1957 & 0.2366 & 0.0254 \end{bmatrix}$$

Subsequent to this stage, the process of centrality ranking is accomplished by using Equation 5.3 on the created adjacency matrix. The obtained ranking results allow for the identification of nodes with the highest centrality scores, which serve as indicators of their relative importance in the graph. These scores are used in the final stage of our pipeline, namely the summary generation, which leverages the top-ranked nodes to produce the summary.

5.3 Experimental Study

In this section we assess the performance of NoWRANK for document summarisation. We first introduce the datasets that we used, then give our pre-processing and implementation details.

5.3.1 Summarisation Datasets

For the purpose of validating the effectiveness of the proposed method on documents, we conduct experiments on four widely-used datasets gathered from numerous contexts. Table 5.2 shows an overview of the four datasets.

Table 5.2: Statistic of CNN/DM , NYT, PubMed and arXiv datasets.

Datasets	**#docs**	**#avg. doc. word**	**#avg. doc. sent.**	**#avg. sum. word**	**#avg. sum. sent.**
CNN/DM	11490	773.22	33.36	57.75	3.79
NYT	6508	1109.10	32.17	96.31	1.18
PubMed	6658	3142.92	101.60	208.02	7.58
arXiv	6440	6446.10	250.36	166.72	6.22

CNN/DailyMail (CNN/DM) (See et al., 2017) and New York Times (NYT) (Sandhaus, 2008) are the standard single-document datasets with manually-written summaries.

Following Zheng and Lapata (2019), we eliminate documents with summaries shorter than 50 words.

PubMed and arXiv (Cohan et al., 2018) are two large-scale datasets of long and structured scientific articles that uses the abstract section as the ground-truth summary and the long body section as the document. While CNN/DM and NYT contain shorter documents and summaries, PubMed and arXiv are more challenging because they have more rich content and diverse information.

5.3.2 Implementation Details

In our LSA based graph model, in the preprocessing stage, fundamental NLP techniques are implemented, comprising the removal of parenthetical text and the elimination of inconsequential characters, such as the double hyphen (–). In the tokenisation stage, the *sumy.nlp.tokenizers* package from the Summy[3] library is used for defining sentences of the document as tokens. For sentence scores, we updated the *LsaSummarizer* method in the *Summy* library and assigned the obtained values to the node weights.

Our Sentence Feature Scoring-based model applies the same preprocessing procedures as our LSA-based model. Specifically, during the tokenisation phase, we employ *sent_tokenize* from the *NLTK* library (Bird & Loper, 2004). *word_tokenize* from the same library is used to compute sentence length and numerical token calculations, while *pos_tag* from NLTK is used to calculate the Proper Noun ratio. As detailed in Chapter 4.2.2, the resultant values obtained from the sentences are assigned as the node weights for their respective nodes.

[3] https://pypi.org/project/sumy/

In both of our graph models, we construct edges using the same techniques. Building upon our previous research, we assess four distinct models within the $SentenceTransformer$ framework for generating sentence embeddings: *'bert-base-nli-mean-tokens'*, *'roberta-base-nli-stsb-mean-tokens'*, *'distilbert-base-nli-stsb-mean-tokens'*, and *'bert-base-nli-stsb-mean-tokens'*. As illustrated in Table 3.5, we observe slight performance variations among these models and we conduct our experiments with the *'roberta-base-nli-stsb-mean-tokens'*. Using cosine similarity, we evaluate the resulting sentence vectors, as discussed in Section 4.2.4, and assign sentence similarity scores to the margins of the graph. We use the $NetworkX$ package (Hagberg et al., 2008) to create the graph, with edge and node weights specified.

In the last step, we apply Equation 5.5 and generate an adjacency matrix. Next, we rank the sentences using Equation 5.3 and determine the six most important sentences for long documents, and the three most important sentences for short documents that should be included in the summary. This numbers of sentences for the summary were chosen as they are the average numbers sentences in the gold summaries for the long and short documents respectively for our test datasets. For the evaluation stage, the py-rouge package[4] is used to calculate the ROUGE scores.

5.4 Results and Analysis

5.4.1 Automated Evaluation

In Table 5.3 , we compare our approach with previous unsupervised and supervised methods for long document extractive summarisation. In Table 5.4, we also compare the performance with the other commonly used short summarisation datasets.

[4] https://pypi.org/project/py-rouge/

Table 5.3: ROUGE/F1 scores of NoWRANK on the arXiv and PubMed datasets.

		arXiv			**PubMed**		
Category	**Method**	R-1	R-2	R-L	R-1	R-2	R-L
Upper Bound	ORACLE	53.88	23.05	34.90	55.05	27.48	38.66
Baselines	LEAD	33.66	8.94	22.19	35.63	12.28	25.17
	LEXRANK (Erkan & Radev, 2004)	33.85	10.73	28.99	39.19	13.89	34.59
Supervised	SummaRuNNer (Nallapati et al., 2017)	42.81	16.52	28.23	43.89	18.78	30.36
	GlobalLocalCont (Xiao & Carenini, 2019)	43.62	17.36	29.14	44.85	19.70	31.43
	Sent-PTR (Pilault et al., 2020)	42.32	15.63	38.06	43.30	17.92	39.47
Unsupervised	PACSUM (Zheng & Lapata, 2019)	39.33	12.19	34.18	39.79	14.00	36.09
	FAR (Liang et al., 2021)	40.92	**13.75**	35.56	41.98	15.66	37.58
	HIPORANK (Dong et al., 2021)	39.34	12.56	34.89	43.58	**17.00**	39.31
Current Work	NoWRANK$_{LSA}$	**43.05**	12.98	**39.27**	**44.05**	15.53	**41.92**
	NoWRANK$_{Sentence\_Feature}$	**42.33**	12.73	**40.54**	**44.27**	15.72	**43.51**

* Results are taken from Dong et al., 2021; Liang et al., 2021. Underlined values indicate the highest values in supervised methods, and bold values indicate the highest values in unsupervised methods.

Table 5.4: ROUGE/F1 scores of NoWRANK on the CNN/DM and NYT datasets.

		CNN/DM			**NYT**		
Category	**Method**	R-1	R-2	R-L	R-1	R-2	R-L
Upper Bound	ORACLE	52.59	17.62	36.67	61.63	41.54	58.11
Baselines	LEAD-3	40.49	17.66	36.75	35.50	17.20	32.00
	LEXRANK (Erkan & Radev, 2004)	34.68	12.82	31.12	30.75	10.49	26.58
Supervised	EXTRACTION (J. Xu & Durrett, 2019)	40.70	18.00	36.80	44.30	25.50	37.10
	REFRESH (Narayan et al., 2018a)	41.30	18.40	35.70	41.30	22.00	37.80
	BertExt (Y. Liu & Lapata, 2019)	43.25	20.24	39.63	-	-	-
Unsupervised	PACSUM (Zheng & Lapata, 2019)	40.70	17.80	36.90	41.40	21.70	37.50
	FAR (Liang et al., 2021)	40.83	17.85	36.91	41.61	**21.88**	37.59
	Liu et al. (J. Liu et al., 2021)	**41.60**	**18.50**	37.80	42.20	21.80	**38.20**
Current Work	NoWRANK$_{LSA}$	39.45	14.02	**39.13**	**44.03**	20.32	36.41
	NoWRANK$_{Sentence\_Feature}$	39.94	14.42	**39.58**	**45.19**	21.60	37.75

* Results are taken from Dong et al., 2021; Liang et al., 2021. Underlined values indicate the highest values in supervised methods, and bold values indicate the highest values in unsupervised methods.

Upper bound and baseline techniques are included in the first block of both tables. ORACLE (Nallapati et al., 2017) used a greedy algorithm to generate an oracle summary for each document (Zheng & Lapata, 2019). The algorithm explores different combinations of sentences and generates an oracle consisting of multiple sentences which maximise the ROUGE score against the gold summary. LEAD and LEXRANK (Erkan & Radev, 2004) are strong unsupervised baselines. LEAD extracts the document's first sentences to provide a summary. In the second block of each table, there are supervised neural extractive summarisation methods. We compare our method with SummaRuNNer (Nallapati et al., 2017), GlobalLocalCont (Xiao & Carenini, 2019), Sent-PTR (Pilault et al., 2020) in Table 5.3, EXTRACTION J. Xu and Durrett, 2019 , REFRESH (Narayan et al., 2018a), BertExt (J. Liu et al., 2021) in Table 5.4. Graph-Based unsupervised extractive summarisation methods are presented in the third blocks of each table. In the last blocks, we show the performance of NoWRANK.

As seen in Table 5.3, our two methods outperform all the other unsupervised graph-based methods by wide margins in terms of R-1,R-L F1 scores ($NoWRANK_{LSA}$ arXiV: R-1 +2.13, R-L +3.71 PubMed: R-1 +0.47, R-L +2.61 ; $NoWRANK_{SentenceFeature}$ arXiV: R-1 +1.41, R-L +4.98 PubMed: R-1 +0.69, R-L +4.2).In addition, the R-L score is higher than the supervised methods in both datasets ($NoWRANK_{LSA}$ arXiV: R-L +1.21 PubMed: R-L +2.45 ; $NoWRANK_{SentenceFeature}$ arXiV: R-L +2.48 PubMed: R-L +4.04).

In Table 5.4, the results obtained from the short documents are presented. The findings of our study demonstrate that our techniques exhibit higher performance in the R-1 and R-L metrics when compared to other unsupervised methodologies ($NoWRANK_{LSA}$ CNN/DM: R-L +1.33 NYT: R-1 +1.83; $NoWRANK_{SentenceFeature}$ CNN/DM: R-L +1.78 NYT: R-1 +2.99). It should be noted, however, that the observed performance advantage, while noteworthy, is not substantial enough to yield a meaningful difference in short documents when contrasted with greater differences observed for longer documents.

5.4.2 Sentence Position Distribution

The present study seeks to conduct a performance evaluation of NoWRANK on the PubMed validation dataset by comparing the position distribution of extracted sentences for PACSUM, HIPORANK, and ORACLE. A visual representation of the position distribution of extracted sentences is provided in Figure 5.5.

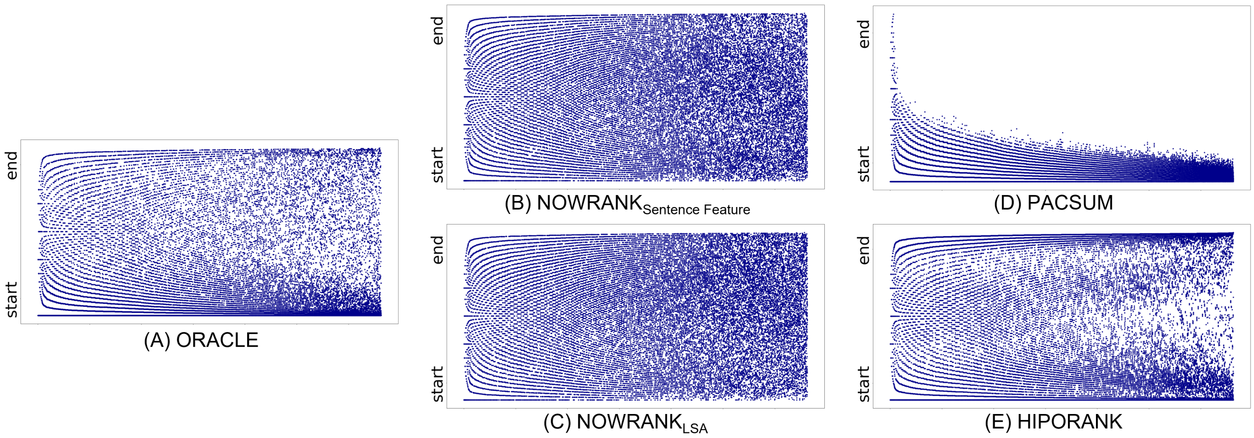


Figure 5.5: The sentence position distribution of extracted sentences by different models on the PubMed validation set. On the x-axis, documents are ordered by article length, from shortest to longest.

The analysis results presented in Figure 5.5 demonstrate that ORACLE selects sentences from short documents in a consistent manner throughout the document. However, for longer documents, there is a stronger emphasis on selecting sentences from the beginning and ending sections, while still maintaining consistent selection from the middle portions. In summary, ORACLE's sentence selection process varies depending on the length of the document, with more focus on the beginning and ending sections for longer documents. As the document length increases, the sentence distribution of both NoWRANK-generated summarization systems remains similar to that of ORACLE, indicating a homogenous distribution of sentence selection. As a difference, NoWRANK's sentence selection process continues to exhibit a homogeneous distribution as the document length increases.

When we examine the PACSUM, PACSUM primarily selects sentences from the earlier sections of the document and this tendency this becomes more apparent the longer the document. When compared to ORACLE, PACSUM exhibit comparable distributions for only a few short documents. The similarity between the sentence position distributions of our proposed models and ORACLE suggests that our models are better suited for summarizing documents of varying lengths than PACSUM.

When considering HIPORANK, it is observed that it exhibits a sentence distribution more similar to ORACLE than PACSUM. However, for longer documents, HIPORANK places greater emphasis on the beginning and ending sections than ORACLE. Comparing HIPORANK with NOWRANK, it is evident that NOWRANK stands out with its homogeneous sentence selection process, which remains consistent even as the length of the document increases.

5.5 Discussion

The focus of this research is on extractive summarisation of long documents. We have achieved exceptional performance in summarising long documents especially in RL metric. Although distinct patterns are observed in shorter documents, they also yield commendable results. Upon closer examination, comparable scores are not attainable between CNN/DM and NYT. While there exists a slight difference in RL scores, a notable variance is apparent in R-1 and R-2 scores in these datasets. We hypothesise that this discrepancy may arise from the differing methods used to create the summaries (CNN/DM employs highlighted text from the news to generate the gold summary, while summaries are manually constructed in NYT), or from limitations of the ROUGE metric.

When comparing our approach with ORACLE's sentence distribution, our sentence selection exhibits a relatively homogeneous distribution. This is due to our method being designed to accommodate diverse types of corpora and our system lacking section information, which is a substantial component of scientific documents. Incorporating section information is a parameter that has the potential to enhance summarisation quality in academic publications. However, the vast majority of corpora do not include section information. In future research, we plan to address this limitation by including a section selection stage. For instance, we can focus on crucial sections such as the introduction, conclusion, methodology, and discussion sections of an academic paper while omitting the literature review and background sections from the document.

5.6 Conclusion

In this section, we present an unsupervised graph-based model for extractive long document summarisation and we emphasise the role of nodes in graph-based summarisation. We introduce both a novel node-weighted graph model and a novel centrality ranking algorithm. The effectiveness of the ranking algorithm is measured with two different node-weighted graph models. The proposed method is systematically evaluated on long documents from PubMed and arXiv and short documents from CNN/DM and NYT. Evaluation results show our simple and effective unsupervised approach outperforms previous unsupervised graph-based summarisation models by wide margins, while achieving performance comparable to state-of-the-art supervised models. In the next chapter, the contributions of this theses will be summarised, and potential future works will be discussed.

Chapter 6

Conclusions

The success of supervised methods leads to their increased use across tasks, including natural language processing. However, supervised methods have limitations on long document summarisation tasks. The work described in this thesis integrates pre-trained language models into graph-based methods to overcome limitations with unsupervised methods without depending on gold summaries and domain-specific training documents, while also achieving higher performance.

This work shows that node weights that have existed for decades for graph-based solutions are also useful for document summarisation tasks. Using node-weighted graph models can improve the performance of document summarisation systems by capturing semantic meanings better.

6.1 Contributions

This section highlights the contributions made by this thesis towards improving extractive text summarisation systems. In the following subsections, the contributions of this thesis are recapped in relation to each of the research questions that this thesis seeks to answer.

6.1.1 Semantic Text Similarity

The first problem that this thesis addresses is defining semantic sentence similarity, bearing in mind the existing challenge associated with redundancy removal in long document summarisation.

Research Question 1: How do specific sentence similarity measures affect the redundancy removal process in text summarisation?

The work presented in Chapter 3 concludes that pre-trained language models have an important contribution to make towards sentence similarity. An unsupervised cluster-based enhanced method was developed to demonstrate the effect of sentence similarity in summarisation systems. Ablation studies on this method demonstrate that SentenceBERT-based models offer effective performance, especially in long document summarisation. Moreover, our method shows improved performance (Gokhan et al., 2021) over other unsupervised methods on long documents. However, although the SentenceBERT models provide an effective solution to the sentence similarity problem, a more detailed solution is required to overcome the limitations of our clustering-based method.

6.1.2 A New Graph Model for Summarisation

The second problem is related to the use of graph models that better represent the semantic similarity of the documents and incorporate more information about the documents.

Research Question 2: What advancements in graph models significantly improve the representation of semantic information for text summarisation?

This work provides a novel node-weighted graph model for document summarisation. For the node weights in the graphs, we calculate values representing the significance of the sentences in the document. We construct graphs with undirected edges using SentenceBERT. Experimental results on four benchmark summarisation datasets demonstrate that our method outperforms other previously proposed extractive graph-based unsupervised methods and provides results comparable to those produced using supervised methods (Gokhan et al., 2022).

Our proposed new model in Chapter 4 demonstrates the importance of node weights in graph-based summarisation systems. The basis of this model is to represent the features that will be effective in summarising sentences with node weights and to represent sentence similarities with edges. We assign four sentence feature values to our experiments to the node weights. These values can be increased, or different statistical or semantic values can be represented in node weights. In addition, a similarity measurement method different from models can be applied for edge weights. The hypothesis of this study is that the effect of node weights on summarisation systems should be regarded. Our systematic evaluations on the different datasets prove this hypothesis.

6.1.3 A New Method for Long Document Summarisation

The final research question pertains to integrating graph ranking algorithms through node-weighted graph models for improving long document summarisation system performance.

Research Question 3: How does the implementation of node-weighted graph ranking algorithms specifically impact the summarisation of long documents?

This work provided an unsupervised graph-based model for extractive long document summarisation and we emphasise the role of nodes in graph-based summarisation. We introduced a novel node-weighted graph model and a centrality ranking algorithm. The effectiveness of the ranking algorithm was measured using two different node-weighted graph models. The proposed method is systematically evaluated on long documents from PubMed and arXiv. Evaluation results showed our simple and effective unsupervised approach outperforms previous unsupervised graph-based summarisation models by wide margins, while achieving performance comparable to state-of-the-art supervised models.

The efficient performance of the proposed new ranking method on long document summary is presented in Chapter 5. Furthermore, the developed ranking algorithm for summarisation has the potential for application in various domains beyond summarisation, such as text classification, document clustering, and keyword extraction, among others.

6.2 Limitations and Challenges

This thesis incorporates a series of studies in text summarization, each with its distinct set of challenges and limitations, which are detailed in the respective chapters. However, some overarching challenges impact the general field of study.

6.2.1 Evaluation Challenges in Text Summarization

- **Challenges in applying appropriate statistical methods for model comparison:** Studies developing and comparing text summarization methods do not report sufficient information for their models to compared to newly developed models in the future using the most appropriate statistical methods.

- **Interpreting the size of differences in model performance metrics:** Whilst it is thought that generally a higher average ROUGE-F1 score indicates better model performance than a lower one there is no guidance on how much of a difference is likely required for there to be a material improvement when a model is deployed in practice.

- **Variability in Human Judgement:** Human evaluators are considered the gold standard for summarization assessment, but their judgements can vary significantly, leading to subjective and sometimes inconsistent evaluations.

- **N-Gram Limitations:** Traditional n-gram-based evaluation metrics like ROUGE are limited in their ability to understand the nuanced meaning of language. They cannot adequately capture synonyms, polysemy, or the broader context of sentences, leading to a potentially misleading assessment of content similarity.

- **Semantic Evaluation:** While BERT-based metrics have emerged to better grasp the semantic content of text, they still grapple with the inherent complexity of language and understanding contextually rich or ambiguous content. These metrics are also computationally intensive and may not be feasible for all research contexts.

- **Lack of Standardisation:** There is no universally accepted standard for what constitutes a 'good' summary, leading to a variety of metrics and methods being used, each with its advantages and limitations. This lack of standardisation can make comparisons across studies challenging.

In our research, we primarily used ROUGE due to its widespread adoption and the comparative analysis it allows with other works in literature. However, our study's limitation lies in not incorporating a broader range of evaluation metrics, including more sophisticated statistical analyses and newer semantic understanding methods, which could provide a more nuanced view of summarization performance. The reasons for this include the significant resources required for implementing and interpreting such advanced methods, the complexity of integrating these into the current research framework. While these constraints impacted the breadth of evaluation methods used, they also reflect common limitations faced by researchers in the field, emphasizing the need for accessible and efficient evaluation tools that can provide deep insights into summarization performance.

6.2.2 Recent Developments in Natural Language Processing

NLP has experienced rapid advancements, significantly affecting various methodologies. While these developments present exciting opportunities, they also introduce certain challenges and limitations for existing methods.

The advent of large language models like ChatGPT has shifted the landscape of NLP, setting new standards for understanding and generating text. These models, trained on extensive and diverse data, have achieved remarkable fluency and context understanding, which can be particularly challenging for more traditional methods to match. Our unsupervised graph-based approach, while effective in many aspects, might not capture the nuanced language understanding and generation capabilities exhibited by such models.

Additionally, the flexibility and adaptability of models like ChatGPT, where style, tone, and focus can be adjusted with relative ease, present a challenge for more rigid approaches. Our method's focus on structure and relationships within the text, crucial for summarization, might not fully encapsulate the stylistic or tonal nuances sometimes necessary for high-quality summaries.

Another consideration is the rate of development in NLP. As new models and techniques continually emerge, keeping methodologies updated and ensuring they remain at the forefront of the field can be a significant challenge. The research and techniques that are state-of-the-art today might quickly become outdated, necessitating continual adaptation and learning.

However, it's also important to note the distinct advantages our approach brings to the table. The unsupervised nature of our method makes it valuable for scenarios where labelled data is scarce or unavailable. Its focus on the structural representation of text provides a clear, interpretable mechanism for summarization, beneficial for certain types of analysis and applications.

In conclusion, while recent developments in NLP provide exciting new directions and capabilities, they also pose challenges and limitations for existing unsupervised graph-based methods. Recognising these factors is crucial in guiding future research and development in the field, ensuring that methodologies not only leverage the strengths of recent advancements but also address their limitations.

6.3 Future Work

Although we have addressed the research questions of this study, there are possible lines of research that can be further investigated in the future.

6.3.1 A Hybrid Approach for Summarisation

In our current researches, we conduct a series of experiments aimed at measuring the similarities among sentences in each procured summary. These experiments involve the application of SentenceBERT and Cosine similarity, as presented in Section 4.2.4, to compute the similarity scores between summary sentences. We then evaluate the mean of sentence similarity scores in the summary by proportioning it to the length of summaries. Our sample dataset consists of 100 documents each taken from the CNN/DM, NYT, arXiv and PubMed datasets. We subsequently generate summaries of these documents using the GUSUM and NoWRANK methods, as well as the K-Means Clustering approach. To assess the performance of these methods, we compare the similarities of the resultant summaries with the gold summaries.

Table 6.1: Similarity scores of summaries on 100 documents selected from of each dataset.

	CNN/DM	**NYT**	**arXiv**	**PubMed**
Gold Summary	0.468	0.402	0.542	0.515
K-Means Clustering	0.416	0.417	0.490	0.447
GUSUM	0.660	0.716	0.650	0.609
NoWRANK_SentenceFeatures	0.727	0.757	0.751	0.734
NoWRANK_LSA	0.722	0.753	0.745	0.737

Table 6.1 illustrates that the sentences included in the summary from the cluster-based approach exhibit similarity scores comparable to those in the gold summaries, thereby establishing the effectiveness of clustering-based approaches in addressing redundancy removal, which is a crucial aspect of text summarisation. In contrast, our graph-based methods yield higher similarity scores; however, the measurements conducted in this thesis demonstrate the proficiency of graph-based approaches in identifying significant sentences, another key aspect of summarisation. These study results suggest that there is potential for a hybrid model that integrates both clustering and graph-based techniques to overcome the limitations associated with each approach. As such, future researchers could look to leverage these findings to enhance the quality of summarisation systems.

6.3.2 Multi-Document Summarisation

Future work should explore the use of graph-based methods for multi-document summarisation. Researchers in this area still face the challenge of better identifying the most salient documents and sentences. For multi-document summarisation systems, a pipeline in which similar documents or phrases are first clustered and then evaluated using our node-weighted graph models could be developed in the future.

6.3.3 Human Evaluation Dataset for Document Summarisation

Evaluating the system's performance is a significant challenge for text summarisation. Despite the availability of numerous automated assessment tools, human evaluation is generally considerable to be the most effective approach for measuring the performance of summarisation systems. Researchers develop their own set of questions to test their methods. However, these question sets are often not shared, which makes it challenging to compare the systems. Preparing and sharing validated summarisation benchmark question-answer sets, which are extensively used in the summarisation field, would substantially help future research.

6.3.4 Application of NoWRANK in Different Domains

Graph-based methods are used for many tasks in natural language processing. In future studies, the applicability of the new method we proposed in this study could also be investigated in NLP tasks such as question-answering, information retrieval, sentiment analysis etc. Moreover, the potential for integration of node-weighted ranking algorithms into decision support, social network analysis and drug delivery systems other than NLP could also be explored.

References

Achananuparp, P., Hu, X., & Shen, X. (2008). The evaluation of sentence similarity measures. In I.-Y. Song, J. Eder, & T. M. Nguyen (Eds.), *Data warehousing and knowledge discovery* (pp. 305–316). Springer. https://doi.org/10.1007/978-3-540-85836-2_29

Agarwal, N., Gvr, K., Reddy, R. S., & Rosé, C. P. (2011). Scisumm: A multi-document summarization system for scientific articles. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, 115–120.

Alguliyev, R., Aliguliyev, R., Isazade, N., Abdi, A., & Idris, N. (2019). Cosum: Text summarization based on clustering and optimization. *Expert Systems*, *36*. https://doi.org/10.1111/exsy.12340

Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: A review. *Artificial Intelligence Review*, 1–81.

Babar, S., & Thorat, S. (2014). Improving text summarization using fuzzy logic & latent semantic analysis. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, *1*(4), 170–177.

Bae, S., Kim, T., Kim, J., & Lee, S.-g. (2019). Summary level training of sentence rewriting for abstractive summarization. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 10–20. https://doi.org/10.18653/v1/D19-5402

Baldeon Suarez, J., Martínez, P., & Martínez, J. L. (2020). Combining financial word embeddings and knowledge-based features for financial text summarization UC3M-MC system at FNS-2020. *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, 112–117. https://aclanthology.org/2020.fnp-1.19

Basu Roy Chowdhury, S., Zhao, C., & Chaturvedi, S. (2022). Unsupervised extractive opinion summarization using sparse coding. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1209–1225. https://doi.org/10.18653/v1/2022.acl-long.86

Bichi, A. A., Keikhosrokiani, P., Hassan, R., & Almekhlafi, K. (2022). Graph-based extractive text summarization models: A systematic review [Publisher: Faculty of Management, University of Tehran]. *Journal of Information Technology Management*, *14*, 184–202. https://doi.org/10.22059/jitm.2022.84899

Bird, S., & Loper, E. (2004). NLTK: The natural language toolkit. *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 214–217. https://aclanthology.org/P04-3031

Bookstein, A., Klein, S. T., & Raita, T. (1995). Detecting content-bearing words by serial clustering. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 319–327.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine [Proceedings of the Seventh International World Wide Web Conference]. *Computer Networks and ISDN Systems*, *30*(1), 107–117. https://doi.org/https://doi.org/10.1016/S0169-7552(98)00110-X

Cagliero, L., Garza, P., & Baralis, E. (2019). Elsa: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis. *ACM Transactions on Information Systems (TOIS)*, *37*(2), 1–33.

Celikyilmaz, A., Bosselut, A., He, X., & Choi, Y. (2018). Deep communicating agents for abstractive summarization. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1662–1675. https://doi.org/10.18653/v1/N18-1150

Celikyilmaz, A., Clark, E., & Gao, J. (2020). Evaluation of text generation: A survey. *ArXiv*, *abs/2006.14799*.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 1–14. https://doi.org/10.18653/v1/S17-2001

Chandrasekaran, D., & Mago, V. (2021). Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, *54*(2). https://doi.org/10.1145/3440755

Chen, J., & Zhuge, H. (2014). Summarization of scientific documents by detecting common facts in citations [Special Section: The Management of Cloud Systems, Special Section: Cyber-Physical Society and Special Section: Special Issue on Exploiting Semantic Technologies with Particularization on Linked Data over Grid and Cloud Architectures]. *Future Generation Computer Systems*, *32*, 246–252. https://doi.org/https://doi.org/10.1016/j.future.2013.07.018

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 615–621. https://doi.org/10.18653/v1/N18-2097

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 670–680. https://doi.org/10.18653/v1/D17-1070

De Boni, M., & Manandhar, S. (2003). The use of sentence similarity as a semantic relevance metric for question answering. *New Directions in Question Answering*, 138–144.

Deutsch, D., & Roth, D. (2022). Benchmarking answer verification methods for question answering-based summarization evaluation metrics. *Findings of the Association for Computational Linguistics: ACL 2022*, 3759–3765. https://doi.org/10.18653/v1/2022.findings-acl.296

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dong, Y., Mircea, A., & Cheung, J. C. K. (2021). Discourse-Aware Unsupervised Summarization for Long Scientific Documents. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1089–1102. https://doi.org/10.18653/v1/2021.eacl-main.93

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology*, *38*(1), 188–230. https://doi.org/https://doi.org/10.1002/aris.1440380105

El-Haj, M., AbuRa'ed, A., Litvak, M., Pittaras, N., & Giannakopoulos, G. (2020). The financial narrative summarisation shared task (FNS 2020). *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, 1–12. https://aclanthology.org/2020.fnp-1.1

El-Haj, M., Zmandar, N., Rayson, P., AbuRa'ed, A., Litvak, M., Pittaras, N., & Giannakopoulos, G. (2021). The Financial Narrative Summarisation Shared Task (FNS 2021). *The Third Financial Narrative Processing Workshop (FNP 2021)*.

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679. https://doi.org/https://doi.org/10.1016/j.eswa.2020.113679

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457–479.

Farouk, M. (2019). Measuring sentences similarity: A survey. *Indian Journal of Science and Technology*, *12*. https://doi.org/10.17485/ijst/2019/v12i25/143977

Farouk, M., Ishizuka, M., & Bollegala, D. (2019). Graph matching based semantic search engine. In E. Garoufallou, F. Sartori, R. Siatri, & M. Zervas (Eds.), *Metadata and semantic research* (pp. 89–100). Springer International Publishing.

Faudree, R. (2003). Graph theory. In R. A. Meyers (Ed.), *Encyclopedia of physical science and technology (third edition)* (Third Edition, pp. 15–31). Academic Press. https://doi.org/https://doi.org/10.1016/B0-12-227410-5/00296-9

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 1606–1611.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, *47*(1), 1–66.

Gehrmann, S., Deng, Y., & Rush, A. (2018). Bottom-up abstractive summarization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4098–4109. https://doi.org/10.18653/v1/D18-1443

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*(3), 245–288. https://doi.org/10.1162/089120102760275983

Gokhan, T., Smith, P., & Lee, M. (2021). Extractive financial narrative summarisation using SentenceBERT based clustering. *Proceedings of the 3rd Financial Narrative Processing Workshop*, 94–98. https://aclanthology.org/2021.fnp-1.18

Gokhan, T., Smith, P., & Lee, M. (2022). GUSUM: Graph-based unsupervised summarization using sentence features scoring and sentence-BERT. *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, 44–53. https://aclanthology.org/2022.textgraphs-1.5

Gokhan, T., Smith, P., & Lee, M. (2023). Node-weighted centrality ranking for unsupervised long document summarization. In E. Métais, F. Meziane, V. Sugumaran, W. Manning, & S. Reiff-Marganiec (Eds.), *Natural language processing and information systems* (pp. 299–312). Springer Nature Switzerland.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 19–25.

Grusky, M., Naaman, M., & Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 708–719. https://doi.org/10.18653/v1/N18-1065

Gupta, H., & Patel, M. (2021). Method of text summarization using lsa and sentence based topic modelling with bert. *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 511–517.

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15).

Haider, M. M., Hossin, M. A., Mahi, H. R., & Arif, H. (2020). Automatic text summarization using gensim word2vec and k-means clustering algorithm. *2020 IEEE Region 10*

Symposium (TENSYMP), 283–286. https://doi.org/10.1109/TENSYMP50017.2020.9230670

Hardy, H., Narayan, S., & Vlachos, A. (2019). HighRES: Highlight-based reference-less evaluation of summarization. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3381–3392. https://doi.org/10.18653/v1/P19-1330

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 1693–1701.

Iskender, N., Polzehl, T., & Möller, S. (2021). Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, 86–96. https://aclanthology.org/2021.humeval-1.10

John, A., Premjith, P., & Wilscy, M. (2017). Extractive multi-document summarization using population-based multicriteria optimization. *Expert Systems with Applications*, *86*, 385–397.

Khan, R., Qian, Y., & Naeem, S. (2019). Extractive based text summarization using kmeans and tf-idf. *International Journal of Information Engineering and Electronic Business*, *11*, 33–44. https://doi.org/10.5815/ijieeb.2019.03.05

Kim, B., Kim, H., & Kim, G. (2019). Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. *NAACL-HLT*.

Koh, H. Y., Ju, J., Liu, M., & Pan, S. (2022). An empirical survey on long document summarization: Datasets, models and metrics [Just Accepted]. *ACM Comput. Surv.* https://doi.org/10.1145/3545176

Kornilova, A., & Eidelman, V. (2019). BillSum: A corpus for automatic summarization of US legislation. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 48–56. https://doi.org/10.18653/v1/D19-5406

Koupaee, M., & Wang, W. Y. (2018). Wikihow: A large scale text summarization dataset.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., & Socher, R. (2019). Neural text summarization: A critical evaluation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 540–551. https://doi.org/10.18653/v1/D19-1051

Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9332–9346. https://doi.org/10.18653/v1/2020.emnlp-main.750

Kumar, A., & Jain, R. (2015). Sentiment analysis and feedback evaluation. *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 433–436. https://doi.org/10.1109/MITE.2015.7375359

La Quatra, M., & Cagliero, L. (2020). End-to-end training for financial report summarization. *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, 118–123. https://aclanthology.org/2020.fnp-1.20

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259–284.

Lehmann, E. L., Romano, J. P., & Casella, G. (1986). *Testing statistical hypotheses* (Vol. 3). Springer.

Liang, X., Wu, S., Li, M., & Li, Z. (2021). Improving unsupervised extractive summarization with facet-aware modeling. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1685–1697. https://doi.org/10.18653/v1/2021.findings-acl.147

Likas, A., Vlassis, N., & J. Verbeek, J. (2003). The global k-means clustering algorithm [Biometrics]. *Pattern Recognition*, *36*(2), 451–461. https://doi.org/https://doi.org/10.1016/S0031-3203(02)00060-2

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81. https://aclanthology.org/W04-1013

Litvak, M., Last, M., & Friedman, M. (2010). A new approach to improving multilingual summarization using a genetic algorithm. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 927–936. https://aclanthology.org/P10-1095

Liu, J., Hughes, D. J. D., & Yang, Y. (2021). Unsupervised extractive text summarization with distance-augmented sentence graphs. In *Proceedings of the 44th international acm sigir conference on research and development in information retrieval* (pp. 2313–2317). Association for Computing Machinery. https://doi.org/10.1145/3404835.3463111

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3730–3740. https://doi.org/10.18653/v1/D19-1387

Louis, A., & Nenkova, A. (2013). Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, *39*(2), 267–300. https://doi.org/10.1162/COLI_a_00123

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, *2*(2), 159–165.

Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-based text summarization using modified textrank. In J. Nayak, A. Abraham, B. M. Krishna, G. T. Chandra Sekhar, & A. K. Das (Eds.), *Soft computing in data analytics* (pp. 137–146). Springer Singapore.

Manakul, P., & Gales, M. (2021). Long-span summarization via local attention and content selection. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language*

Processing (Volume 1: Long Papers), 6026–6041. https://doi.org/10.18653/v1/2021.acl-long.470

Maybury, M. T. (1995). Generating summaries from event data [Summarizing Text]. *Information Processing & Management*, *31*(5), 735–751. https://doi.org/https://doi.org/10.1016/0306-4573(95)00025-C

McKeown, K. R., Klavans, J. L., Hatzivassiloglou, V., Barzilay, R., & Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, 453–460.

Meng, R., Thaker, K., Zhang, L., Dong, Y., Yuan, X., Wang, T., & He, D. (2021). Bringing structure into summaries: A faceted summarization dataset for long scientific documents. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 1080–1089. https://doi.org/10.18653/v1/2021.acl-short.137

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. https://aclanthology.org/W04-3252

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3075–3081.

Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 280–290. https://doi.org/10.18653/v1/K16-1028

Narayan, S., Cohen, S. B., & Lapata, M. (2018a). Ranking sentences for extractive summarization with reinforcement learning. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1747–1759. https://doi.org/10.18653/v1/N18-1158

Narayan, S., Cohen, S. B., & Lapata, M. (2018b). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1797–1807. https://doi.org/10.18653/v1/D18-1206

Nazari, N., & Mahdavi, M. A. (2019). A survey on automatic text summarization. *Journal of AI and Data Mining*, *7*(1), 121–135. https://doi.org/10.22044/jadm.2018.6139.1726

Nenkova, A., Maskey, S., & Liu, Y. (2011). Automatic summarization. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 3. https://aclanthology.org/P11-5003

Nie, P., Zhang, J., Li, J. J., Mooney, R., & Gligoric, M. (2022). Impact of evaluation methodologies on code summarization. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4936–4960. https://doi.org/10.18653/v1/2022.acl-long.339

Ouyang, Y., Li, W., Wei, F., & Lu, Q. (2009). Learning similarity functions in graph-based document summarization. *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages. Language Technology for the Knowledge-Based Economy*, 189–200. https://doi.org/10.1007/978-3-642-00831-3_18

Owczarzak, K., Conroy, J. M., Dang, H. T., & Nenkova, A. (2012, June). An assessment of the accuracy of automatic evaluation in summarization. In J. M. Conroy, H. T. Dang, A. Nenkova, & K. Owczarzak (Eds.), *Proceedings of workshop on evaluation metrics and system comparison for automatic summarization* (pp. 1–9). Association for Computational Linguistics. https://aclanthology.org/W12-2601

Ozsoy, M., Cicekli, I., & Alpaslan, F. (2010). Text summarization of turkish texts using latent semantic analysis. *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, 869–876.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Pawar, S., Manjula Gururaj, H., & Chiplunar, N. N. (2022). Text summarization using document and sentence clustering [4th International Conference on Innovative Data Communication Technology and Application]. *Procedia Computer Science*, *215*, 361–369. https://doi.org/https://doi.org/10.1016/j.procs.2022.12.038

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. http://www.aclweb.org/anthology/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Peyrard, M., Botschen, T., & Gurevych, I. (2017). Learning to score system summaries for better content selection evaluation. *Proceedings of the Workshop on New Frontiers in Summarization*, 74–84. https://doi.org/10.18653/v1/W17-4510

Pilault, J., Li, R., Subramanian, S., & Pal, C. (2020). On extractive and abstractive neural document summarization with transformer language models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9308–9319. https://doi.org/10.18653/v1/2020.emnlp-main.748

Qazvinian, V., & Radev, D. R. (2008). Scientific paper summarization using citation summary networks. *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 689–696. https://aclanthology.org/C08-1087

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics*, *28*(4), 399–408. https://doi.org/10.1162/089120102762671927

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Rahimi, S. R., Mozhdehi, A. T., & Abdolahi, M. (2017). An overview on extractive text summarization. *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 0054–0062.

Ramirez-Orta, J., & Milios, E. (2021). Unsupervised document summarization using pre-trained sentence embeddings and graph centrality. *Proceedings of the Second Workshop on Scholarly Document Processing*, 110–115. https://doi.org/10.18653/v1/2021.sdp-1.14

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. https://doi.org/10.18653/v1/D19-1410

Ross, S. M. (2010a). Chapter 1 - introduction to statistics. In S. M. Ross (Ed.), *Introductory statistics (third edition)* (Third Edition, pp. 1–15). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-374388-6.00001-6

Ross, S. M. (2010b). Chapter 9 - testing statistical hypotheses. In S. M. Ross (Ed.), *Introductory statistics (third edition)* (Third Edition, pp. 387–442). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-374388-6.00009-0

Ruhnau, B. (2000). Eigenvector-centrality — a node-centrality? *Social Networks*, *22*(4), 357–365. https://doi.org/https://doi.org/10.1016/S0378-8733(00)00031-9

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Sandhaus, E. (2008). The New York Times Annotated Corpus LDC2008T19. *Linguistic Data Consortium, Philadelphia*. https://catalog.ldc.upenn.edu/LDC2008T19

Sarkar, K., Saraf, K., & Ghosh, A. (2015). Improving graph based multidocument text summarization using an enhanced sentence similarity measure. *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, 359–365. https://doi.org/10.1109/ReTIS.2015.7232905

Schluter, N. (2017). The limits of automatic summarisation according to ROUGE. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 41–45. https://aclanthology.org/E17-2007

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083. https://doi.org/10.18653/v1/P17-1099

Sharma, E., Li, C., & Wang, L. (2019). BIGPATENT: A large-scale dataset for abstractive and coherent summarization. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2204–2213. https://doi.org/10.18653/v1/P19-1212

Shirwandkar, N. S., & Kulkarni, S. (2018). Extractive text summarization using deep learning. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–5. https://doi.org/10.1109/ICCUBEA.2018.8697465

Singh, A. (2020). PoinT-5: Pointer network and T-5 based financial narrative summarisation. *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, 105–111. https://aclanthology.org/2020.fnp-1.18

Singh, A., Singh, R. R., & Iyengar, S. R. S. (2020). Node-weighted centrality: a new way of centrality hybridization. *Computational Social Networks*, *7*(1), 6. https://doi.org/10.1186/s40649-020-00081-w

Song, K., Li, C., Wang, X., Yu, D., & Liu, F. (2022). Towards abstractive grounded summarization of podcast transcripts. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4407–4418. https://doi.org/10.18653/v1/2022.acl-long.302

Steen, J., & Markert, K. (2021, April). How to evaluate a summarizer: Study design and statistical analysis for manual linguistic quality evaluation. In P. Merlo, J. Tiedemann, & R. Tsarfaty (Eds.), *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 1861–1875). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.160

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, *4*, 93–100.

Suanmali, L., Binwahlan, M. S., & Salim, N. (2009). Sentence features fusion for text summarization using fuzzy logic. *2009 Ninth International Conference on Hybrid Intelligent Systems*, *1*, 142–146. https://doi.org/10.1109/HIS.2009.36

Suanmali, L., Salim, N., & Binwahlan, M. S. (2009). Fuzzy logic based method for improving text summarization. *arXiv preprint arXiv:0906.4690*.

Tandel, A., Modi, B., Gupta, P., Wagle, S., & Khedkar, S. (2016). Multi-document text summarization - a survey, 331–334. https://doi.org/10.1109/SAPIENCE.2016.7684115

Vilca, G. C. V., & Cabezudo, M. A. S. (2017). A study of abstractive summarization using semantic representations and discourse level information. *International Conference on Text, Speech, and Dialogue*, 482–490.

Wang, Z., Ma, L., & Zhang, Y. (2016). A novel method for document summarization using word2vec. *2016 IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, 523–529. https://doi.org/10.1109/ICCI-CC.2016.7862087

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., . . . Rush, A. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Xiao, W., & Carenini, G. (2019). Extractive summarization of long documents by combining global and local context. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3011–3021. https://doi.org/10.18653/v1/D19-1298

Xu, J., & Durrett, G. (2019). Neural extractive text summarization with syntactic compression. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3292–3303. https://doi.org/10.18653/v1/D19-1324

Xu, S., Zhang, X., Wu, Y., Wei, F., & Zhou, M. (2020). Unsupervised extractive summarization by pre-training hierarchical transformers. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1784–1795. https://doi.org/10.18653/v1/2020.findings-emnlp.161

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, *33*, 17283–17297.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, July). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 11328–11339, Vol. 119). PMLR. https://proceedings.mlr.press/v119/zhang20ae.html

Zhang, Y., Ni, A., Mao, Z., Wu, C. H., Zhu, C., Deb, B., Awadallah, A., Radev, D., & Zhang, R. (2022). Summ^N: A multi-stage summarization framework for long input dialogues and documents. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1592–1604. https://doi.org/10.18653/v1/2022.acl-long.112

Zhao, L., Li, L., & Zheng, X. (2020). A bert based sentiment analysis and key entity detection approach for online financial texts. *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1233–1238.

Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., & Eger, S. (2019). MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 563–578. https://doi.org/10.18653/v1/D19-1053

Zheng, H., & Lapata, M. (2019). Sentence Centrality Revisited for Unsupervised Summarization. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6236–6247. https://doi.org/10.18653/v1/P19-1628

Zhuang, L., Wayne, L., Ya, S., & Jun, Z. (2021). A robustly optimized BERT pre-training approach with post-training. *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, 1218–1227. https://aclanthology.org/2021.ccl-1.108