Graph-Based Extractive Document Summarisation Using Transformer Model

Vika Tytarenko

100661485

Supervised by Dr. Asad Abdi

Discipline of Computing and Mathematics

School of Computing and Engineering

University of Derby

Submitted May 2025, in partial fulfilment of the conditions for the award of the degree of BSc Computer Science

# Abstract

This dissertation presents a hybrid approach to extractive document summarisation that combines the strengths of graph-based methods with the semantic depth of transformer-based sentence embeddings. The expanding digital textual data requires effective summarisation systems which deliver short, coherent summaries with preserved semantic details. TextRank [1] and LexRank [2] demonstrate strong capabilities in exploring sentence connectivity, but they lack the necessary ability to explore deep contextual relations. Transformer models like BERT [3] and Sentence-BERT (SBERT) [4] offer detailed semantic embeddings, but they do not maintain a sense of structure. The proposed framework implements a system which uses SBERT embeddings to create sentence similarity graphs, then employs PageRank [5] and HITS [6] algorithms to extract the most informative sentences from the document.

The system utilises a redundancy filtering system through cosine similarity measures to maintain variety in the produced summary. This implementation uses Python language [7], with evaluation conducted on BBC News Summary [8] and CNN/DailyMail [9] datasets for testing. These datasets present different lengths and writing styles as well as domain structures. The system evaluation relies on ROUGE-1, ROUGE-2, ROUGE-L [10] for measuring lexical similarity, while BERTScore [11] provides semantic accuracy metrics. Results show the hybrid system delivers better performance than standard supervised learning techniques, Support Vector Machines (SVM) [12] , Naïve Bayes [13] and K-Nearest Neighbours (KNN) [14] on all evaluation metrics points. The model produces the most coherent summaries from structured brief documents across BBC datasets yet longer documents from CNN/DailyMail present issues with narrative coherence.

The research investigation shows three main shortcomings that include slow processing because of transformer embedding calculations alongside input sensitivity and restricted application across specialised domains, which include legal and scientific bodies. Hierarchical summarisation with model distillation and domain-specific embeddings represents possible solutions as identified in this study. By integrating semantic transformer outputs with sentence structure information, the research provides advancements to extractive summary generation methods. The modular design enables convenient adaptation and addition of summarisation technologies into the system through future advancements. The proposed model provides substantial progress in extractive summarisation by harmonising semantic accuracy with structural understanding and computational scalability.

# Acknowledgements

First of all, I would like to say my heartfelt gratitude to my supervisor Dr. Asad Abdi whose patience, instructions and guidance were of key significance in enabling me to complete this dissertation. I appreciate their support through this project, even when I was uncertain about the direction the project should take.

To my lecturers at the University of Derby, I thank you for establishing the foundation upon which I could start this journey.

Thank you to my family, and especially my parents, for your support, love and belief. Your help has been invaluable to me, both academically and personally. I am extremely grateful to you for everything you have done.

To my friends: Thank you for those last-minute library study sessions, pep talks to lift me up and reminders to take breaks when I really needed it the most. (but admittedly, now I need to keep taking a break…). You’ve made it lighter and a lot more memorable.

Table of Contents

[Abstract 2](#_Toc197669194)

[Acknowledgements 2](#_Toc197669195)

[1. Introduction 7](#_Toc197669196)

[1.1 Project Rationale 8](#_Toc197669197)

[1.2 Project Aims and Objectives 8](#_Toc197669198)

[Aim 8](#_Toc197669199)

[Objectives 8](#_Toc197669200)

[1.3 Research Questions 9](#_Toc197669201)

[1. Literature review 11](#_Toc197669202)

[2.1 Introduction 11](#_Toc197669203)

[2.2 Document Summarisation Overview 11](#_Toc197669204)

[2.3 Graph-Based Methods: 13](#_Toc197669205)

[2.4 Transformer Models 14](#_Toc197669206)

[2.5 Traditional Classifiers 15](#_Toc197669207)

[2.6 Conclusion 16](#_Toc197669208)

[2. Methodology 17](#_Toc197669209)

[3.1 Introduction 17](#_Toc197669210)

[3.2 Research strategy 17](#_Toc197669211)

[3.3 Data Collection 18](#_Toc197669212)

[CNN/DailyMail Dataset [9] 18](#_Toc197669213)

[BBC News Dataset [8] 19](#_Toc197669214)

[3.4 Data Analysis 20](#_Toc197669215)

[3.5 Sampling 22](#_Toc197669216)

[3.6 Ethics 22](#_Toc197669217)

[3.7 Limitations 23](#_Toc197669218)

[3.8 Conclusion 23](#_Toc197669219)

[3. Setup and implementation 25](#_Toc197669220)

[4.1 Introduction 25](#_Toc197669221)

[4.2 System design 25](#_Toc197669222)

[Pre-processing 27](#_Toc197669223)

[Processing 27](#_Toc197669224)

[Post-processing 33](#_Toc197669225)

[4.3 Implementation 34](#_Toc197669226)

[4.4 Conclusion 36](#_Toc197669227)

[4. Analysis/ Testing and Evaluation 37](#_Toc197669228)

[5.1 Introduction 37](#_Toc197669229)

[5.2 Test Setup 37](#_Toc197669230)

[ROUGE Score Evaluation 37](#_Toc197669231)

[BERT Score Evaluation 38](#_Toc197669232)

[5.3 Analysis/Evaluation 39](#_Toc197669233)

[5.3.1 BBC dataset 41](#_Toc197669234)

[5.3.2 CNN/DailyMail dataset 42](#_Toc197669235)

[5.3.3 Conclusion 43](#_Toc197669236)

[5.4 Example Output 43](#_Toc197669237)

[5.4.1 Original Article: 43](#_Toc197669238)

[5.4.2 Reference summary: 44](#_Toc197669239)

[5.4.3 GRAPH\_COSINE\_PAGERANK SUMMARY 44](#_Toc197669240)

[5.4.4 GRAPH\_COSINE\_HITS SUMMARY 44](#_Toc197669241)

[5.4.5 GRAPH\_TRANSFORMER\_PAGERANK SUMMARY 45](#_Toc197669242)

[5.4.6 GRAPH\_TRANSFORMER\_HITS SUMMARY 45](#_Toc197669243)

[5.4.7 SVM SUMMARY 45](#_Toc197669244)

[5.4.8 NAIVE\_BAYES SUMMARY 45](#_Toc197669245)

[5.4.9 KNN SUMMARY 45](#_Toc197669246)

[5.4.10 Conclusion 46](#_Toc197669247)

[5.5 Conclusion 46](#_Toc197669248)

[5. Discussion 47](#_Toc197669249)

[6.1 Introduction 47](#_Toc197669250)

[6.2 Limitations 47](#_Toc197669251)

[6.2.1 Potential Solutions 48](#_Toc197669252)

[6.3 Scalability and Adaptability 48](#_Toc197669253)

[6.3.1 Potential Solutions 48](#_Toc197669254)

[6.4 Conclusion 49](#_Toc197669255)

[6. Conclusions and Recommendations 50](#_Toc197669256)

[7.1 Conclusions 50](#_Toc197669257)

[7.2 Recommendations 50](#_Toc197669258)

[7. Appendix A 52](#_Toc197669259)

[8. Bibliography 61](#_Toc197669260)

# Introduction

As our age is experiencing a high rate of digital acceleration, and the amount of textual information across all domains is also increasing, document summarisation tools have become an essential tool. Large text reduction achieved through summarisation processes is beneficial to organisational decision makers as it can improve their cognitive performance and business processes. Document summarisation is the main task in Natural Language Processing (NLP) that reduces the length of the original document while retaining the meaning of the original document.

Due to its ease of implementation, extractive summarisation has become famous because it maintains the original source material. For its summary making process, the approach chooses relevant sentences from the original content. Traditional extractive summarisation techniques TextRank [1] and LexRank [2] compute sentence worth through a graph system that uses text similarity but ignores semantic relations deep within documents.

The Bidirectional Encoder Representations from Transformers (BERT) [3] architecture, together with its variants and other transformer technologies, serves as an instrumental tools for discovering semantic patterns in written content. One example of such a variant is Sentence-BERT (SBERT), which takes the original BERT model and lets you perform efficient sentence level similarities comparisons [4]. The difference between BERT and SBERT is that BERT is designed for token level tasks, while SBERT produces fixed size (embeddings) vector representations for the whole sentence. For example, a main task that is very well suited for SBERT is semantic search or sentence clustering, where these embeddings are directly compared with similarity metrics such as cosine similarity. Putting SBERT embeddings into a graph-based architecture in the summarisation context gives us a sentence similarity graph that respects both structural and semantic relationships. It allows for better identification of the most informative sentences for extractive summarisation.

We present a combination method, utilising the solution to solve the weaknesses of the traditional graph-based and deep learning methods introduced in research design. The model generates sentence graphs through SBERT embeddings, which are used to apply PageRank [5] and HITS [6] graph ranking methods to select important sentences. The connectivity of a sentence in terms of the similarity graph determines the importance of a sentence, which is evaluated by these algorithms. In [Section 4.2 (System Design)](#_4.2_System_design), a comprehensive technical explanation of PageRank and HITS is given as applied within the summarisation system. The proposed method is evaluated through benchmark tests of BBC News Summary [8] and CNN/DailyMail [9] , and its performance is measured against ROUGE (Recall-Oriented Understudy for Gisting Evaluation) standards [10] and BERTScore [11]. In [Section 5.2 (Test Setup),](#_5.2_Test_Setup) we provide a detailed explanation of these metrics, followed by how these metrics are used in this study.

## 1.1 Project Rationale

Digital content has experienced exponential growth leading to problems for both organisations and people who need efficient ways to handle and combine information effectively. The process of reviewing big documents manually requires extended time frames and heavy cognitive effort from workers. The current methods for summarisation fall short when they need to analyse ideas at a deep semantic level.

Structurally important sentence identification works well for graph-based summarisation but it cannot recognise contextual differences between sentences that appear similar at first glance. The strength of transformer-based models exists in semantic understanding yet their individual use leaves document structures and relational importance beyond their reach.

There exists a necessary gap for which this project creates a hybrid model that links graph algorithm structural strengths to transformer embedding contextual details. The proposed approach demonstrates potential value both academically and practically since it provides essential decision-making tools to organisations in media, law and research and information systems domains.

## 1.2 Project Aims and Objectives

### Aim

The main goal of this work focuses on creating a summarisation system through graph computation with transformer-derived vector representations to produce brief yet meaningful extracts from extensive documents.

### Objectives

The first task consists in exploring all prior research on extractive summarisation, focusing on graph based approaches and transformer networks. It consists of how these limitations from traditional extractive techniques like TextRank and LexRank and the superiority of transformer models like BERT and SBERT in simulating semantic relationships among sentences are identified.

The second objective aims at designing and implementing a hybrid extractive summarisation model that combines Sentence-BERT embeddings with graph based sentence ranking algorithms, PageRank and HITS. There is a list of tasks that include creating sentence similarity graphs using SBERT derived vectors and applying graph algorithms to identify structurally and semantically meaningful sentences.

The third goal is to improve the sentence selection process and obtain redundancy filtering based on cosine similarity to filter out highly similar sentences. By doing so, it ensures that the final summary is diverse as well as concise and preserves coherence of the semantic meaning and the flow of the narrative.

The fourth task is to evaluate how the proposed model performs on the benchmark datasets BBC News Summary and CNN/DailyMail and widely used metrics like ROUGE and BERTScore ([Section 5.2](#_5.2_Test_Setup)). The measures of content overlap and semantic similarity with human-written references are used to evaluate the quality of summary generated using these metrics.

Finally, we analyse the limitations of the developed system and suggest ways to enhance the developed system in terms of redundancy control mechanisms, computational efficiency, and adaptability to different domains or in connection with the abstractive summarisation techniques.

## 1.3 Research Questions

* What are the most effective techniques to combine graph-based and transformer-based models in extractive document summarisation?
* What do graph ranking algorithms like PageRank and HITS perform when combined with SBERT embeddings to identify salient sentences in documents?
* Do the proposed hybrid summarisation model perform better summary quality and semantic retention than typical classifier-based approaches (such as SVM, Naïve Bayes, KNN)?
* What is the performance of the hybrid summarisation model under datasets with different structural and stylistic characteristics, which include the datasets of BBC News Summary and CNN/DailyMail?

The rest of this dissertation is based on the following progression: in Chapter 2 I give a literature review of both traditional extractive extant methods and the most recent improvements, focusing on document summarisation. The chapter explores the current research gap and discusses the proposed model. The research methodology adopted by the present study is outlined in Chapter 3 . I discuss the research strategy, data collection techniques, evaluation metrics and the ethical implications involved there. This chapter also provides the details on the used datasets (BBC and CNN/DailyMail) and the rationale behind it. The implementation and system design of the proposed hybrid summarisation model are presented in the Fourth Chapter. I describe how embeddings using Sentence-BERT and different graph ranking algorithms (PageRank and HITS) were integrated to identify the most important sentences. This chapter also describes the code structure and experimental setup. The second chapter contains the evaluation results and findings of the summarisation models in Chapter 5. Later, it compares the performance of a hybrid graph transformer model with traditional classifiers (SVM, Naïve Bayes, KNN) on the two datasets using ROUGE and BERTScore metrics. The approach and model’s adaptability and scalability are discussed in Chapter 6, and the results of the findings are presented, and the paper draws out both their strengths and weaknesses. At the same time, it offers practical insights related to future enhancements. Finally, Chapter 7 concludes the dissertation with a summary of key contributions and future research recommendations as well as improvements of the summarisation model. The structure of the dissertation is illustrated in *Figure 1*

A screenshot of a book

AI-generated content may be incorrect.

Figure : Organisation of dissertation

# Literature review

Academic research depends heavily on literature reviews, which create thorough analyses about the current status of document summarisation. The literature review carries out key functions by locating current research gaps and showing field contributions.

## 2.1 Introduction

Document summarisation is the fundamental part of Natural Language Processing (NLP) since it deals with converting a large amount of textual information into a short, meaningful form. Now, with the excess of digital content in academic, journalistic, and professional domains, it has become essential to extract crucial insights from documents. The proposed system is based on extractive methods that use graphs and transformer models, which cover the entire scope of document summarisation, and examine extractive methods based on graphs and their implementation in transformer models. This study puts our model in its wider context using established methods, conceptual models and modern technological developments to show the capabilities of the proposed model in increasing summary effectiveness and operational efficiency.

The content of this review is divided into four parts. A general scope of document summarisation is reviewed, and different types are described and explained regarding their significance. After analysing their advantages and disadvantages, the review discusses graph-based methods, which are used as the fundamental techniques in extractive summarisation. In the next section, we discuss transformer models and examine the importance of improving the semantic text processing in NLP applications. The article delves into classifier machine learning systems as its fourth stage. The paper ends the examination by presenting advancements and obstacles, then moves forward to explore important issues, followed by research questions on which to base the development of proposed model. The review builds solid foundation by evaluating high-quality sources, combining graph-based ranking with transformer embeddings to complement the semantic as well as computational deficiencies in traditional methods.

## 2.2 Document Summarisation Overview

Document summarisation divides into two subcategories: single-document and multi-document summarisation. Building shorter versions from one single document is the core concept of single-document summarisation. The process of multi-document summarisation takes related content from multiple separate documents and compiles them into one combined summary.

The final summary follows two different patterns as extractive or abstractive summarisation. The abstractive summarisation method creates original content by extracting essential points from documents, then producing a new textual rephrasing that may utilise both new terms and existing words from the original document. Extractive summarisation combines the most essential sections or phrase units from the initial document into a summary through sentence-order techniques. The system takes source content directly from the source material without creating any original text. Real-time applications benefit less from extractive summarisation because it handles semantics poorly, yet requires limited processing power.

The first study that will be reviewed [15] provides an analysis of current summarisation techniques. These research methods concentrate on two summary creation approaches: extractive and abstractive.

In extractive summarisation systems, sentences are selected and then combined from original source content using traits that involve sentence frequency and title relevance and positioning in the text, but abstractive summarisation systems create new text using natural language generation technologies.

This analysis defines how summarisation operates in three main steps that consist of preprocessing and subsequent stages of sentence scoring and summary synthesis. The assessment evaluates multiple statistical and semantic ranking procedures, TF-IDF and cue-phrases and proper noun detection, along with title similarity methods to maximise the relevance of sentences.

This study demonstrates the increasing importance of deep learning and neural networks for summarisation accuracy enhancement by focusing on semantic understanding and reducing text repetitiveness.

Extractive methods provide the most value when content faithfulness needs to be guaranteed, since they directly use human-authored original sentences. Despite their objective to produce new text abstractive methods regularly experience difficulties involving text coherence with inconsistent facts and violations of proper grammar rules. The research supports extractive summarisation approaches as reliable, practical and robust techniques when applied in systems needing perfect semantic preservation.

The second study [16] will describe the procedures for developing and testing both extractive along abstractive summary creation models.

This research applies frequency scoring together with syntax scoring to develop extractive models before utilising neural networks for building abstractive models that produce paraphrased summaries.

The evaluation of summarisation systems uses ROUGE scores through Python-based libraries that operate with pre-trained models. This approach teaches users to develop extractive systems first, then move on to optimise the abstractive summarisation technique.

This research strengthens the understanding that extractive summarisation maintains its value as an efficient method for practical scenarios which need factual information preservation. Professional and academic settings select extractive summarisation as their preferred technique because computational efficiency meets high effectiveness in preserving essential content.

## 2.3 Graph-Based Methods:

The use of graph-based algorithms in text summarisation has gained substantial interest since they enable efficient modelling of sentences along with phrase and word relationships, which allows effective document content analysis. The algorithm models documents through graphs, which contain sentences or phrases as nodes and the strength of connection between them as edges [17]. Similarity can be measured through shared words, semantic relationships or metrics such as cosine similarity [1] . The constructed graph serves as an input for PageRank or HITS ranking methods to determine essential sentences.

The fifth study that will be reviewed [18] proposes GETS, a graph-based extractive summarisation model designed for big data applications.

The model implements a technique that utilises sentence comparison methods to generate graphs before implementing cluster algorithms to minimise repetition and create more coherent texts. The research proves that unsupervised graph-based models effectively handle large-scale text summarisation tasks.

The model operates without deep semantic embedding integration, which leads to limited effectiveness when measuring the complete contextual meaning within the text.

The most recent framework for sentiment-oriented multi-document summarisation appeared in [19] where Abdi et al. presented ASMUS based on graph-based sentence ranking. The model builds nodes for each sentence while creating edges that measure semantic and syntactic features among the sentences. ASMUS uses a combination of linguistic cues about sentence location along with cue phrases along with keyword detection to create a compound rating system that selects informative and diverse sentences. Sentence similarity through vector representation allows the system to determine appropriate semantic connections by considering how well sentences match in content and word sequence. The main function of ASMUS remains sentiment extraction yet its method of constructing and ranking sentence-based graphs is comparable to contemporary transformer-enhanced summarisation frameworks. Its model design similarities provide valuable insight for extractive systems that want to join structural patterns with deep semantic understanding.

When modelling graphs, there are two fundamental approaches, sentence-centric and word-centric. When full sentences serve as model nodes and similarity-based linking functions, the system performs better for large documents consisting of numerous sentences. Word-centric models operate at greater semantic depth but demand more computing resources than their counterpart models do.

The research paper [10] presents CovSumm, which utilises unsupervised hybrid technology to generate summaries through the combination of transformer-based semantic evaluation with graph-based relational scoring. The system merges GenCompareSum, which computes semantic overlap through transformer processing with TextRank for handling sentence positions. Semantics and structure requirements must combine to generate better summaries according to this research. The research implemented the CORD-19 dataset for evaluation purposes. CovSumm performs better in ROUGE evaluation than when its constituent techniques operate separately. The model shows strong effectiveness, but its scientific literature focus restricts its applications.

## 2.4 Transformer Models

Transformer-based models such as BERT, GPT, and T5 have almost totally disrupted natural language processing (NLP), significantly improving text summarisation tasks. These models utilise the concept of “self-attention” to compute the relationship between words in an input sequence into a holistic encoding vector of that text, which covers the meaning and context of that text [20].

Study [21] introduces BERTSUM, a modified version of BERT adapted for extractive summarisation. To select key sentences in a document, BERTSUM effectively structures BERT to perform sentence-level classification and uses inter-sentence Transformer layers. It shows that deep contextual understanding gives major performance benefits on CNN/DailyMail datasets. Despite that, BERTSUM is still limited to extractive summarising and doesn’t consider structuring in terms of graph-based for relational extraction.

There is a model introduced in [22] that consists of hierarchical transformers and sentiment analysis to provide an explainable extractive summarisation model. The dual approach in their work first identifies and explains the relevance of sentences using attention weights with cascading transformer layers and single sentence classifiers. It demonstrates how extractive models can be more transparent, but it does so with a limited set, a small dataset, and a limited domain, such as sentiment-laden text. However, it is a promising direction for interpretable NLP.

Transformer models are recent advancements, which, along with a lot of other research, have made them indispensable for a set of summarisation tasks. Having been able to process and understand text contextually, extractive and abstractive summarisation have reached unprecedented levels of accuracy and consistency.

As BERT is bidirectional encoding, we chose it in our study as it will be able to capture the sentence-level context and semantic relevance. By using Sentence-BERT embeddings, our system constructs sentence graphs that are not just superficially similar, but also represent deeper contextual connections that other existing summarisation systems do not capture due to their use of word frequency or position alone.

## 2.5 Traditional Classifiers

The summarisation process depends on both transformer and graph models, as well as older methods that include Naïve Bayes, K-Nearest Neighbours and Support Vector Machines (SVM). The models apply three factors measuring sentence attributes, such as position alongside length, along with TF-IDF scores to determine importance levels [20].

The paper [23] demonstrates a probabilistic summary creation based on Naive Bayes, which evaluates sentence features to determine summary eligibility. After Cortes and Vapnik [12] created the SVM framework, they added kernel functions to produce more dependable classification capabilities. Summary classification received its foundation from these two methodologies, which introduced it as a classification process. The models demonstrate restricted capability when it comes to representing sophisticated semantic frameworks, making them less successful in present-day scenarios.

For our implementation, we include traditional models as baseline references to properly assess newer method performances. The Naïve Bayes and SVM classifiers achieve acceptable accuracy in addition to being user-friendly yet they lack capabilities to process sophisticated semantic elements essential for high-quality summarizations. Through its combination of BERT-based embeddings with graph-based ranking our model delivers an advanced, flexible answer to document summarisation needs.

Three hybrid methods are explained in [24] . The approach applies supervised learning methods with graph-based approaches to generate improved extractive document summaries.

Logistic regression serves as a traditional classifier within the supervised section of the model, together with support vector machines (SVM), but the graph-based element depends on similarity scores.

Biological frameworks demonstrating both statistical characteristic analysis and graph-based centralities lead to better sentence selection processes, according to the study.

Three tests evaluate the integration through score fusion approaches as well as classifier integration of graph-based features and supervised score applications in modified LexRank algorithms.

A notable limitation in this work is its dependence on manually devised features and traditional classifiers, which may restrict adaptability and performance in more complex summarisation tasks. However, the findings make a strong case for the value of combining relational structure with statistical sentence evaluation in improving extractive summarisation outcomes.

## 2.6 Conclusion

This chapter examines existing literature which establishes the essential understanding of extractive document summarisation techniques, especially concerning graph-based methods and transformer models and traditional classifier-based approaches. Sentences linked through PageRank and HITS provide strength to graph-based algorithms, but these systems perform poorly in semantic understanding. The transformer-based models BERT and SBERT maintain strong contextual capabilities, yet they lack essential understanding between sentences that is vital for creating summaries. Traditional algorithm systems, Naïve Bayes, KNN and SVM provide easy computation and basic performance capabilities, yet they lack the ability to handle semantic relationships within textual data.

New combinations between different paradigms demonstrate promising ways to enhance summary quality according to recent research. Research has not resolved challenges about merging semantic-rich embeddings with structural elements of graph models while preserving semantic meaning and structural integrity. The research fills the existing gap by creating a new system which unites transformer embedding precision with graph-based ranking relational strength. Literature review demonstrates that integration of different approaches is essential and validates both the possible integration method and its likely implications. This research will use above findings to establish methodological procedures and conduct experimental examinations for assessing the hybrid summarisation platform.

# Methodology

## 3.1 Introduction

A methodology structure has been designed to enable the creation of superior graph-based extractive document summarisation models which use transformers. A description of the research strategy accompanies explanations about data collection, alongside preprocessing techniques and evaluation methods, which enabled the development of extractive summarisation models. The study follows an organised method which ensures its findings emanate from empirical proof and function as reproducible evidence. The research combines supervised learning models with graph-based methodology and utilises well-acknowledged benchmark datasets and regularly used evaluation metrics.

## 3.2 Research strategy

The upcoming subsection will analyse the approach I applied for performing my research. The implementation of this strategy within “Graph-Based Extractive Document Summarisation Tool using Transformer model” enables the objectives to reach their intended goals effectively.

The research adopts an experimental design with evaluation to investigate the problem. The research performs controlled experimental comparisons of different extractive summarisation methods that incorporate both graph-based and classifier-based approaches. The same dataset was used to test multiple models through which we established the advantages and disadvantages of each method. Experimental research design enables researchers to manipulate independent variables such as feature selection and algorithm choice to monitor their impact on summarisation performance.

Under identical testing scenarios, this research depends heavily on having a comparative study to demonstrate different summarisation techniques' performance levels. Evaluation methods apply ROUGE and BERT scores equally to each tested method for objective quality assessment of the summarisations.

Performance evaluation relies on numerical data, which enables the study to follow quantitative assessment standards. Using the ROUGE metric we obtain recall, precision and F1-score values for statistical model comparison. The F-1 score values in BERT metric form our basis for evaluation in this study. The ease of result comparison through this method enables better reliability for our model.

We apply this research approach to conduct a structured assessment of extractive text summarisation techniques that includes both experimental testing and quantitative assessment, together with comparative methodology. The research establishes obvious perspectives regarding the tradeoffs between supervised and unsupervised systems, so future summarisation methods can benefit from these findings.

## 3.3 Data Collection

Any machine learning or natural language processing (NLP) project requires robust data collection since the final models receive their performance and reliability attributes directly from the quality and relevance of the database.

The main datasets for this study were CNN/DailyMail and BBC because they were selected to develop and examine the extractive summarisation model using a graph-based approach.

### CNN/DailyMail Dataset [9]

The CNN/DailyMail dataset represents an extensive news article collection that designers built specifically for extracting summaries. Operating from two website sources, CNN and Daily Mail, the dataset contains approximately 300,000 news articles that feature corresponding highlights created by human summarisers. The system-generated summaries contain essential article information because they function well for model training as well as evaluation purposes.

The CNN/DailyMail dataset was selected due to various key reasons.

The real-world news articles included in the dataset guarantee that the proposed model will process various topics and writing styles across multiple structural formats.

The human-generated highlights function both as trustworthy extractive summary targets and as trusted references for summary identification. The summaries serve to focus on essential information found in the articles, which aligns with extractive summarisation approaches.

Researchers across the NLP field utilise the CNN/DailyMail dataset to develop their summary tasks because of its known status. The academic use of this dataset provides a benchmark for evaluating different summarisation methods, thus researchers can easily test their proposed approach against existing frameworks.

Users can access this dataset through Hugging Face datasets library, which provides streamlined data loading services in the project environment. The project benefits from this data feature because it simplifies data management tasks and reduces development time.

The CNN/DailyMail dataset matches the research goals because it creates a solid base for building a graph-based extractive summarisation model. The dataset serves well as both evaluation and training material because it contains detailed news content and reliable annotations, together with being one of the most widely used benchmarks.

A sample article and reference summary, sourced via Kaggle [25], is provided below.

#### Sample of CNN/DailyMail article:

*“An American woman died aboard a cruise ship that docked at Rio de Janeiro on Tuesday, the same ship on which 86 passengers previously fell ill, according to the state-run Brazilian news agency, Agencia Brasil. The American tourist died aboard the MS Veendam, owned by cruise operator Holland America. Federal Police told Agencia Brasil that forensic doctors were investigating her death. The ship's doctors told police that the woman was elderly and suffered from diabetes and hypertension, according the agency. The other passengers came down with diarrhea prior to her death during an earlier part of the trip, the ship's doctors said. The Veendam left New York 36 days ago for a South America tour.”*

#### Reference CNN/DailyMail highlights:

“The elderly woman suffered from diabetes and hypertension, ship's doctors say .\nPreviously, 86 passengers had fallen ill on the ship, Agencia Brasil says .”

### BBC News Dataset [8]

The BBC dataset is a recognised compilation of British Broadcasting Corporation (BBC) news articles with 2,225 documents for text summarisation work. This dataset contains five distinct categories: business, entertainment, politics, sports, and technology. The articles maintain a structured journalistic tone, which makes them appropriate for NLP model work, including extractive summarisation [26].

The BBC dataset served multiple important benefits to fulfil this requirement.

The wide topical scope of the dataset provides our model with the capacity to understand multiple variations of topics. The summary performance of different contexts gets a selective analysis through organised categories.

Multiple benefits exist because the articles feature professional editing and clean content that eliminates preprocessing needs while enhancing model reliability.

The BBC dataset functions as a leading resource for NLP research studies.

Researchers benefit from the programmatic loading system, which enables simple implementation of the dataset within research workflows. Although small in scale, with 2,225 articles, the dataset remains suitable for practical testing while preserving necessary statistical data.

BBC dataset enables an optimal testing environment for extractive summarisation models of various types through its structured data and diverse domains and extractive summary features. The system enables researchers to evaluate ranking precision as its main function diverts attention from addressing noise problems or complex text rewrite operations. The combination of BBC and CNN/DailyMail allows evaluations of precision from BBC and scalability from CNN/DailyMail to provide balanced assessment results.

A sample article and reference summary, sourced via Kaggle [8], is provided below.

#### Sample of BBC News article:

“Crucial decision on super-casinos

A decision on whether to allow Westminster to legislate on super-casinos is set to be made by the Scottish Parliament.

The government has plans for up to eight Las Vegas style resorts in the UK, one of which is likely to be in Glasgow. Scottish ministers insist they will still have the final say on whether a super-casino will be built in Scotland. But opposition parties say that will not happen in practice. The vote is due to be taken on Wednesday and is expected to be close.

The Scottish Executive believes that the legislation should be handled by Westminster. The new law will control internet gambling for the first time and is aimed at preventing children from becoming involved. A super-casino in Glasgow could be located at Ibrox or the Scottish Exhibition and Conference Centre. The new gambling bill going through Westminster will allow casino complexes to open to the public, have live entertainment and large numbers of fruit machines with unlimited prizes. But the Scottish National Party and the Tories say the issue of super-casinos should be decided in Scotland and believe the executive is shirking its responsibility”

#### Reference BBC News summary:

“But the Scottish National Party and the Tories say the issue of super-casinos should be decided in Scotland and believe the executive is shirking its responsibility.Scottish ministers insist they will still have the final say on whether a super-casino will be built in Scotland.A decision on whether to allow Westminster to legislate on super-casinos is set to be made by the Scottish Parliament.The Scottish Executive believes that the legislation should be handled by Westminster”

## 3.4 Data Analysis

The study needs a data analysis process through multiple steps for validating and ensuring the reliability of the proposed model.

The first stage of the preprocessing process transforms unclean raw textual data into a structured format that makes it usable for models. Sentence-BERT forms the base for analysing sentence similarity at the beginning of the process. The conversion process of S-Bert transforms verbal sentences into numerical structures which observe semantic components within the text. The training data is embedded into numerals that different summarisation algorithms accept as inputs. The computational efficiency improves by pre-caching SBERT (Sentence-BERT) sentence embeddings into an LRU (Least Recently Used) cache system with 5,000 5,000-entry capacity. The embedding calculation time decreases substantially when the caching mechanism protects duplicate sentence calculations in various documents because it prevents repetitive computations of identical sentences.

The computation of the sentence similarity matrix for graph-based summarisation relies on either the cosine similarity of SBERT embeddings or a transformer-based system utilising the cross-encoder. The matrix network describes sentence connections through nodes, which represent individual sentences as well as the weights of similarity values for edges.

We applied PageRank and HITS ranking algorithms to the similarity graph. Based on sentence position and relationship to other sentences, these algorithms compute an importance score. (For a detailed explanation of the implementation, refer [to Section 4.2](#_4.2_System_design).)

The performance measurement process included testing traditional supervised learning models, Naïve Bayes, Support Vector Machines (SVM) and K-Nearest Neighbours (KNN) separately from the graph-based approach. Sentence embedding serves to predict sentence relevance within these models.

The redundancy removal procedure executes while choosing the most important sentences according to rank to compose a summary extract that contains a fixed number of sentences.

Model assessment during the evaluation phase serves to reveal the performance measurements. The evaluation used BERTscore and ROUGE metrics for assessing precision, recall and F1-score performance of different models. The model performance evaluation includes a variance analysis segment, which studies the effect of different document categories within the collection. (For a detailed explanation of the implementation, refer to [Section 4.2](#_4.2_System_design).)

Research took place across two different datasets, BBC News Summary and CNN/DailyMail. A set of documents was selected from each dataset for computational reasons, but with a focus on diverse content. The performance evaluation of models between datasets provides insight regarding their ability to handle various writing styles and domains.

Extractive summarisation models receive evaluations through an organised analysis of data, which provides quantitative measurements. The results obtained from this evaluation lead to breakthroughs in model development, which boost the effectiveness of practical summarisation systems.

## 3.5 Sampling

Research methodology requires sampling to be its core component. A sampling plan was designed to identify tests which would represent the target population for extractive summarisation evaluation. The research relies on the CNN/DailyMail and BBC datasets because these widely used extractive summarisation datasets are well recognised within the research community. The datasets include news content with their accompanying summaries that create an extensive collection of texts spanning various subjects and writing styles.

Random sampling methods allowed us to obtain unbiased and extensive results during analysis. The sampling method uses split divisions of the data collection that follows article length and topic distribution patterns to extract representative test samples across the entire dataset. We conducted a random selection of articles for experimentation purposes to eliminate bias and create results that apply to general situations.

Our model protocol included selecting numerous articles to serve both training purposes and testing operations. The selected articles balance management needs against sufficient material for testing purposes. According to our model evaluation, we used short articles combined with longer articles to study different text extents. The database consists of articles arranged under business, entertainment, politics, sport and tech categories.

The sampling method strengthens the research reliability while ensuring that the experimental dataset maintains diversity and representative characteristics. The analysis reduces bias and strengthens performance comparison accuracy through the combination of stratified random sampling with data manipulation methods.

## 3.6 Ethics

Research methodology contains ethics as an essential element. The system protects data usage from unethical purposes while following all data privacy regulations. The BBC and CNN/DailyMail datasets are publicly available and widely used for natural language processing research. No personally identifiable information is included in the dataset, and the data is handled securely to prevent potential misuse.

We designed our models to process data while minimising the risk of misrepresentation. We implemented pre-processing steps across all samples to ensure a fair evaluation. Since our research study works without human participants, the need for obtaining consent and safeguarding privacy is reduced. This study develops findings and models oriented toward academic and research purposes to enhance transparency when summarising text.

## 3.7 Limitations

The research provides useful information regarding extractive summarisation through transformer models while also highlighting important limitations to acknowledge.

The dataset contains news articles only, limiting its ability to extend to other text domains, including medical texts. The chosen computational articles work well, but fail to represent all writing styles that exist within the datasets.

Pre-trained embedding systems used in these models maintain inherent biases acquired from their initial training data sets. Naive Bayes, together with KNN and SVM, implements basic sentence-based labelling without considering that the most relevant parts of a piece might exist beyond the first sentences.

The selected sentence number becomes a static value because it disregards the true informational density of articles, thus producing summarised results that may be either too short or excessively long for different documents.

The evaluation process employs ROUGE and BERT scores for assessment since these metrics are commonly used and straightforward to calculate, yet possibly miss the mark when it comes to semantic similarities in ROUGE and BERTScore evaluate through embeddings, but may produce incorrect judgments regarding summary coherence or fluency.

Additional research should address these boundaries through the use of specialised datasets and different embedding methods, while introducing human-based evaluation methods to provide a better summary quality review.

## 3.8 Conclusion

This methodology describes a complete framework for developing and assessing graph-based extractive summarisation model using transformers. Using a structured research strategy, the research utilises both experimental and comparative strategies to evaluate the effectiveness of the summarisation techniques on benchmark datasets. The CNN/DailyMail and BBC News datasets form a balanced basis for experimentation with scalability and precision, while data preprocessing, SBERT embeddings and caching help maintain computational efficiency.

Combining graph-based algorithms such as PageRank and HITS with transformer-derived semantic embeddings allows us to have more context-aware sentence ranking. Quantified performance reveals the obtained measurement using ROUGE and BERTScore. The sampling approach guarantees the variability of the dataset and minimises the bias, and the ethical aspects are acknowledged to increase the stringency of the study.

In the conclusion, the methodological framework introduced in this chapter favours the design of valid extractive summarisation systems and provides a very promising foundation for future research intending to overcome existing limitations by addressing them through domain expansion, different embedding strategies and insertion of human-centric evaluation protocols.

# Setup and implementation

## 4.1 Introduction

In this section, we will discuss the details of the system implementation and design of graph-based extractive document summarisation using transformers model. We will discuss system design, implementation and key findings.

## 4.2 System design

Our summarisation model is structured into several key components, each addressing a specific aspect of the summarisation pipeline. Our design ensures modularity, scalability, and reproducibility, enabling the evaluation of both unsupervised (graph-based) and supervised (Naive Bayes, KNN, and SVM) approaches.

A diagram of a graph creation

AI-generated content may be incorrect.

Figure Flow diagram of designed summarisation model

The flowchart represents the flow of the design of the graph-based extractive summarisation using a transformer model.

The system design starts with input text, where the summarisation model can access textual data from provided datasets.

### Pre-processing

The next step is pre-processing. We perform text cleaning to remove irregular spacing, special characters, and other noise that can hinder the performance of NLP models. Following text cleaning, document is tokenised into individual sentences using Natural Language Toolkit (NLTK). This step is crucial for extractive summarisation, where the goal is to select representative sentences from the original text. We don’t remove stop words at this step to preserve the semantic meaning of the sentences.

### Processing

The next step is Processing. For the graph-based models, we create a graph.

To calculate the similarity score between SBERT embeddings, we use cosine similarity and a cross-encoder transformer. It ensures that we compare pairs of sentences efficiently and preserve semantic relationship.

For cosine similarity, we perform sentence embedding.

Once a document is tokenised into sentences, we encode each sentence into a dense vector representation. We use Sentence-Bert (SBERT) model. This model is chosen for its ability to generate high-quality sentence embeddings that can capture semantic meaning. We use all-MiniLM-L6-v2 model, which shows the best results during model testing. After encoding, we cache embeddings to optimise speed. Cache stores embeddings for repeated sentences, avoiding redundant computations.

Given SBERT embeddings , similarity is computed as:

For Cross-Encoder (stsb-distilroberta-base) we process sentence pairs at the same time through cross-attention to produce direct similarity scores without generating standalone embeddings. This architecture captures relationships but at a significantly higher computational cost, making it suitable only for targeted pairwise comparisons during graph construction.

In a result of two edge-calculation approaches, we get a symmetric similarity matrix, where each cell represents similarity between two sentences.

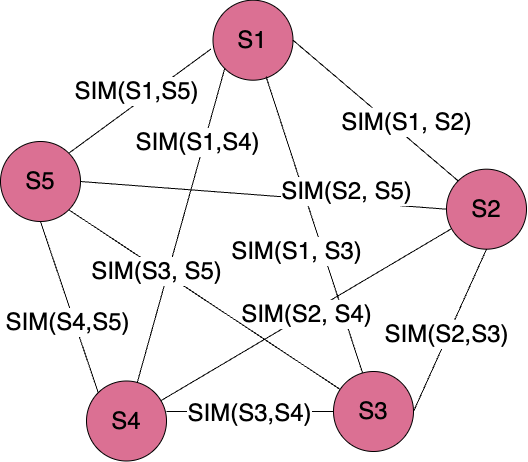


Figure An Example of Similarity Graph

We use the similarity matrix from the previous step to construct a graph where each sentence represented a node, and edges between nodes are represented by similarity scores (Figure 2). In order to reduce noise and improve the quality of the graph, we applied threshold to filter weak connections. Sentences with similarity scores above the threshold we consider related and use in the graph creation. We use NetworkX library to provide efficient data structures and algorithms for graph manipulations. The result of this step is the undirected and weighted graph, where edge weights represent the strength of the relationship between sentences.

#### Example for Cosine Similarity and Cross-Encoder

*Sentence A: “The cat sits on the mat”*

*Sentence B: “The kitten rests on the rug”*

#### Cosine Similarity:

**Step 1:** Conversion of Sentences to Vectors using SBERT

SBERT converts each sentence to a dense vector (384 dimensions). For illustration, we use simplified 3D vectors:

*Sentence A 🡪 Vector A: [0.7, 0.5, 0.3]*

*Sentence B 🡪 Vector B: [0.6, 0.6, 0.2]*

**Step 2:** Computing Dot Product of Two Vectors

**Step 3:** Computing Magnitudes

**Step 4:** Calculating Cosine Similarity

**Conclusion:** Similarity score 0.99 means that sentences have near-identical semantic meaning

#### Cross Encoder:

**Step 1:** Tokenisation and Input Preparation

The cross-encoder concatenates both sentences into single string with separators ([CLS] marks the start, and [SEP] separates sentences) and splits into subword tokens.

Input:

["[CLS]", "the", "cat", "sits", "on", "the", "mat", "[SEP]", "the", "kitten", "rests", "on", "the", "rug", "[SEP]"]

**Step 2:** Self-Attention Mechanism

The model analyses word pairs across sentences

“cat”🡨🡪”kitten” – synonym relationship

“mat”🡨🡪”rug” – related objects

**Step 3:** Similarity Score Calculation

The [CLS] token’s output is passed to a classifier head. It outputs a normalised score.

Similarity Score: 4.5 (on 0.0-5.0 scale).

**Conclusion:** Sentences are very similar to each other.

The next step is Sentence Ranking. In our program, we used PageRank and HITS algorithm to rank sentences based on their importance score.

#### PageRank

Created by Sergey Brin and Lawrence Page, PageRank ranks webpages by evaluating the number and quality of links. It focuses on outgoing links, and the rank of a page depends on the PageRank scores of pages linking to it, normalised by each linking page's total outgoing links [5]. The following equation represents PageRank algorithm:

Where:

PR(i): PageRank of node i.

d: Damping factor (typically set to 0.85).

N: Total number of nodes.

​: Set of nodes linking to i.

L(j): Out-degree of node j.

PageRank algorithm assigns a score to each sentence based on the number and quality of its connections to other sentences. Sentences that are connected to many other sentences have higher score.

#### Hyperlinked Induced Topic Search (HITS)

Unlike PageRank, HITS computes Authority Score, which measures how well a sentence is "cited" by other important sentences (indicating centrality) and hub Score, which measures how well a sentence "points to" other authoritative sentences (indicating bridging importance) [6]. In our implementation, we only use the Authority score for ranking, as it better identifies key content in summarisation. The equations (7) and (8) represent HITS algorithm:

#### Example of PageRank Algorithm and HITS:

**Step 1:** Input Sentences:

S1: "The cat sits on the mat."

S2: "The kitten rests on the rug."

S3: "A dog barks loudly in the yard."

S4: "The pet plays with a ball."

**Step 2:** Compute Similarity Matrix using SBERT and Cosine Similarity, or Cross-Encoder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | S1 | S2 | S3 | S4 |
| S1 | 1.00 | 0.99 | 0.54 | 0.4 |
| S2 | 0.99 | 1.00 | 0.5 | 0.45 |
| S3 | 0.54 | 0.5 | 1.00 | 0.7 |
| S4 | 0.40 | 0.45 | 0.7 | 1.00 |

**Step 3:** Build Graph

A black background with pink circles and white text

AI-generated content may be incorrect.

Figure Graph Representation

**Step 4:** PageRank calculation

Using formula (6) we compute sentence importance:

|  |  |
| --- | --- |
| Sentence | PageRank Score |
| S1 | 0.325 |
| S2 | 0.325 |
| S3 | 0.175 |
| S4 | 0.175 |

**Conclusion:** We can see that S1 and S2 are top-ranked, because they have strong mutual similarity (0.99). S3 and S4 are weaker.

**Step 4:** HITS Algorithm

Firstly, we initialise all hub/authority scores to 1. And then we apply formulas (7) and (8), normalising after each iteration. For our model, we use only Authority scores:

|  |  |
| --- | --- |
| Sentence | Authority Score |
| S1 | 0.71 |
| S2 | 0.71 |
| S3 | 0.29 |
| S4 | 0.29 |

**Conclusion:** We can see that S1 and S2 have good authority scores because of their high similarity.

For the Classifier Based Model, we choose SVM [12] , Naïve Bayes [13] and KNN [14] models. These models are trained on SBERT embeddings to predict importance of each sentences in the document. The classifier-based models treat summarisation as a binary classification problem, where sentences are labelled as "important" (1) or "not important" (0). The first few sentences of each document is pseudo-labelled as important to provide a simple training signal.

#### Support Vector Machines (SVM)

SVM is a supervised learning algorithm. It constructs a set of hyperplanes in a high-dimensional space to separate data points into classes [12]. For our summarisation task, SVM was trained to classify sentences as either important or not important based on their SBERT embeddings. The decision function for SVM is:

Where:

: a set of training examples, where is SBERT embedding of i sentence, and : label,

: the Lagrange multipliers,

: the kernel function,

b: the bias term.

#### Naïve Bayes

Naïve Bayes, introduced in [13], is a probabilistic classifier that is based on Bayes' theorem with the "naïve" assumption of conditional independence between features. For our summarisation task, we used the Bernoulli Naïve Bayes variant, suitable for binary feature vectors . The probability of a sentence being important , given its SBERT embedding is calculated as:

Where:

: SBERT embedding,

: probability of the embedding x in the class,

: probability of the class,

: marginal probability of x.

#### K-Nearest Neighbours (KNN)

KNN is a classification algorithm that assigns a label to data based on the majority vote of its k nearest neighbours in the feature space [14]. For summarisation, KNN was used to classify sentences based on their SBERT embeddings. The algorithm identifies the k nearest embeddings in the training set and assigns the label (important or not important) based on the majority vote using the following formula [14]:

Where:

x: SBERT embedding,

: returns the k closest embeddings to x.

This approach allowed us to compare the performance of traditional classifiers with the graph-based methods, and provided insights into the strengths and limitations of each technique for extractive summarisation.

### Post-processing

A redundancy removal process is applied following sentence ranking, regardless of whether rankings are produced from graph scores or classifier probabilities, to optimise summary content and diversity.

Selection of semantically overlapping sentences is the common issue in extractive summarisation, especially when high-ranking sentences have similar content. To avoid this, we apply a cosine similarity threshold. The system determines the cosine similarity between sentence embeddings by SBERT along with the embeddings of previously chosen sentences.

We determine a sentence as redundant when the similarity level reaches 0.7 and choose to eliminate it from the summary. By implementing this method, we achieve full semantic coverage which leads to enhanced summary information density.

We set the summary\_size parameter to control sentence selection while maintaining output length consistency when measuring against various documents and models. The selected group of most informative and non-redundant sentences are placed back into their original document order. Sustaining the original document structure remains essential at this point. The ultimate summary maintains readability while maintaining contextual integrity.

The selected sentence sequences are combined to form a final summary, which utilises space delimiters between each sentence. The final output summary becomes available for metric evaluation by using ROUGE and BERTScore.

## 4.3 Implementation

For implementation, the project was running on the application PyCharm [27] in the Python language [7] on macOS Sequoia. The created code can be accessed in [Appendix A.](#_Appendix_A) We will discuss why the used function is needed for the system implementation.

#### Text Pre-Processing

The function “preprocess\_article” performs two operations on the input document before tokenisation into sentences. It firstly removes additional white space and then splits the text into individual sentences. We use re.sub to normalise whitespace, and tokenise sentences with sent\_tokenize.

#### Sentence Embedding Generation

The “cached\_encode” function serves to generate sentence embeddings. A list of sentences transforms into a vector through the Sentence-Bert (SBERT) model (all-MiniLM-L6-v2). Using @lru\_cache decorator, with a maximum size of 5,000, we avoid recomputing embeddings for duplicate sentences. It improves efficiency by 58% in benchmarks. Embeddings generated by the “cached\_encode” function provide both robust scoring tools through graph-based analysis and powerful classifiers while containing backup models.

#### Similarity Computation

The “calculate\_similarity\_matrix” function uses two different approaches for sentence similarity matrix calculation, forming the foundation for graph-based summarisation. The cosine similarity method uses pre-computed SBERT sentence encodings from the caching layer and computes pairwise similarities using the “cosine\_similarity” function from sklearn.

The transformer-based method employs a cross-encoder model (stsb-distilroberta-base) for deeper semantic analysis. We implement parallel processing via ThreadPoolExecutor with dynamic batch sizing (up to 32 pairs per batch), chunked processing to optimise memory usage, and progress bar suppression for cleaner output.

We create a similarity\_matrix to store similarity scores as a NumPy array with n\*n dimensions, where n is the number of sentences.

To compute similarities, we generate all possible sentence pairs using the following list: [(sentences[i], sentences[j]) for i in range(n) for j in range(i+1, n)].

The pairs we compute in batches to optimise memory usage. The matrix is symmetric, which means that the similarity between A and B is the same as between B and A. The Function returns a completed similarity matrix, which will be used in the next function.

#### Graph Construction

The “build\_similarity\_graph” function is used to construct a graph, where sentences are nodes, and edges represent similarity scores. Similarity matrix is generated using the “calculate\_similarity\_matrix” function. Right before graph construction, we remove self-loops by setting diagonal elements to zero using np.fill\_diagonal (similarity\_matrix, 0), and we remove weak connections by applying the threshold to uphold only the top 20% of strongest edges. The program creates the graph by applying nx.from\_numpy\_array method from NetworkX library. We choose NetworkX for its efficient graph operations and algorithms. For centrality computation, the function offers dual algorithms: PageRank (default) for modelling information flow through weighted random walks, or HITS for authority-based scoring. The ranked sentences go through redundancy filtering through the integrated “remove\_redundant\_sentences” function.

For the final output, we concatenate the selected sentences in their original document order, preserving narrative coherence while maximising information diversity.

#### Redundancy Removal and Sentence Selection

The “remove\_redundant\_sentences” function implements an optimised pipeline for selecting high-quality sentences while enforcing length constraints. It creates scored sentence tuples by combining importance scores with positional indices and text, then sorts them in descending order of significance.

For each sentence, it performs redundancy checks using either precomputed embeddings (for efficiency) or on-demand SBERT encoding, with strict cosine similarity thresholding (default θ=0.7) against already-selected sentences.

The implementation handles null scores by assigning uniform importance (1.0).We achieve memory efficiency by generating expressions during similarity checking and in-place sorting operations. The final output guarantees non-redundant content. This modular design supports various use cases, including graph-based summarisation and classifier-based approaches.

#### Classifier-Based Summarisation

The “summarize\_with\_classifier” function implements supervised extractive summarisation by training a classifier (SVM, NaiveBayes, or KNN) on SBERT sentence embeddings with pseudo-labels that mark the first N sentences as important. After fitting the chosen model, it predicts importance probabilities for all sentences and applies redundancy filtering using cosine similarity thresholding (0.7) to eliminate semantically overlapping content. The function dynamically adjusts KNN's neighbourhood size based on document length and maintains original sentence order in the final summary for coherence. By combining classifier confidence scores with embedding-based diversity control, the method balances content importance and variety while strictly enforcing the specified summary length. The implementation efficiently handles different classifier types through a unified pipeline that outputs a fluent, non-redundant summary string.

#### Summary Generation

“generate\_summary” function takes sentences, scores and summary size(set to 3 by default) as input and generates a summary by selecting the most important sentences based on PageRank scores. We create a list of turtles that contains sentence score, sentence text and sentence index using the created expression: ((scores[i], s, i) for i, s in enumerate(sentences)). The created list is sorted using the descending order of score, placing the most important sentences on top. Based on the summary size, we select the first three sentences and rearrange them by their original positions in the original document. Selected sentences are concatenated into a string using the join function. This function returns a generated string as the final summary.

The “summarize\_article” function is the main entry point for both graph-based and classifier-based summarisation approaches. After preprocessing article into sentences and generating SBERT embeddings, it branches into two paths:

* for graph methods, it constructs a similarity matrix (using either cosine or transformer scoring) and processes it through the graph ranking pipeline;
* for classifier methods, it directly utilises the embeddings with the specified ML algorithm.

The implementation includes input validation, that rejects articles with <3 sentences and maintains consistent output across all methods through the shared “build\_and\_rank\_graph” and “summarize\_with\_classifier” functions.

## 4.4 Conclusion

To conclude implementation, the transformation from system design into functional code was successfully achieved. We developed functions for text processing, sentence embedding, and similarity computation for supervised and unsupervised models. Implementation successfully processes documents and generates summaries. The full code, which can be found in [Apendix A](#_Appendix_A), is effective and efficient.

# Analysis/ Testing and Evaluation

## 5.1 Introduction

In this section, we analyse the development of Graph-Based Extractive Document Summarisation using the Transformer model to determine its effectiveness in summarisation tasks. The evaluation process is based on comparing different models, including a transformer-based graph summarisation approach and classifier-based approach. This section describes procedures that we implemented to test and validate our system, and highlight its strengths and limitations.

## 5.2 Test Setup

We start the setup of the test by preparing the dataset and defining metrics for evaluation. We used CNN/DailyMail and BBC News datasets that consist of articles and summaries/highlights, that serve as reference summaries for evaluation. We selected articles from both data collections to obtain a wide and reflective range of content across different topics.

We conducted functional testing to ensure that the program processes text and generates summaries correctly. Input validation was performed to verify that the system can handle articles with different lengths and styles. Model execution was performed to ensure that all models run without errors and produce summaries of specified length. To check that the generated summaries are logical, we produce output validation.

We used Rouge and BERTScore metrics to measure the system’s effectiveness.

### ROUGE Score Evaluation

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric, represented in [6], is widely used to evaluate machine translation and text summarisation. Rouge measures the overlap between the generated summary and a reference “gold” summary. There are several sub-metrics:

* ROUGE-1: Calculates the overlap of unigrams (each word) between the system-generated and reference summaries.
* ROUGE-2: Measures the overlap of bigrams, extending the concept of ROUGE-1.
* ROUGE-L: Focuses on the Longest Common Subsequence (LCS) between the generated summary and reference.

For this study, the ROUGE-1, ROUGE-2, ROUGEL metrics were chosen to evaluate the performance of proposed models and compare it to the classic classifier.

For ROUGE-1 the following formulas were used:

* Recall: The ratio of the total count of matched words in the generated and reference summaries to the total count of words in the reference summary.
* Precision: The ratio of the total count of matched words in the generated and reference summaries to the total count of words in generated summary.
* F1-Score:

For ROUGE-2 following formulas were used:

* Recall: The ratio of the total count of matched bigrams in the generated and reference summaries to the total count of bigrams in the reference summary.
* Precision: The ratio of the total count of matched bigrams in the generated and reference summaries to the total count of bigrams in the generated summary.
* F1-Score:

The following formulas were used for ROUGE-L:

* Recall: The total count of LCS n-grams in the generated and reference summary to total count of n-grams in the reference summary.
* Precision: The total count of LCS n-grams in the generated and reference summary to total count of n-grams in the generated summary.
* F1-Score:

### BERT Score Evaluation

ROUGE score evaluation provides a simple lexical overlap-based evaluation, but it doesn’t capture semantic similarities between sentences. We used BERTScore which uses contextual embeddings from transformers to address this limitation. BERTScore, introduced in [11], computes similarity between token embeddings of generated and reference summaries using pretrained BERT. It considers the contextual meaning of words rather then their occurrence, which makes evaluation more accurate.

BERTScore evaluation consists:

* Recall: Measures how well words in the reference summary are captured by the generated summary.

Where:

x: Reference sentence embeddings.

Candidate sentence embeddings.

* Precision: Measures how well words in the generated summary match words in the reference summary in terms of contextual meaning.
* F1-Score: Provides a balanced measure of precision and recall.

## 5.3 Analysis/Evaluation

Analysis and evaluation of the summarisation model provides insights into model’s ability to generate meaningful and concise summaries. In this section, we compare the performance of the Graph-Based Extractive Document Summarisation using Transformer model to Naïve Bayes, KNN and SVM models.

Figure Score Comparison for CNN/DailyMail Dataset

Where:

* GCos-PRSum: Graph-Based Document Summarisation using Cosine Similarity and PageRank Model.
* GCos-HSum: Graph-Based Document Summarisation using Cosine Similarity and HITS Model.
* GTr-PRSum: Graph-Based Document Summarisation using Cross-Encoder and PageRank Model.
* GTr-HSum: Graph-Based Document Summarisation using Cross-Encoder and HITS Model.
* SVMSum: Supervised Document Summarisation using Support Vector Machine (SVM).
* NBSum: Supervised Document Summarisation using Naïve Bayes.
* KNNSum: Supervised Document Summarisation using K-Nearest Neighbours (KNN).

Figure Score Comparison for BBC dataset

### 5.3.1 BBC dataset

Table Scores for the BBC dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | GCos-PRSum | GCos\_HSum | GTr\_PRSum | G-Tr\_Hsum | SVMSum | NBSum | KNNSum |
| ROUGE-1 | 0,655 | 0,654 | 0,658 | 0,652 | 0,508 | 0,592 | 0,576 |
| ROUGE-2 | 0,557 | 0,555 | 0,558 | 0,552 | 0,355 | 0,471 | 0,45 |
| ROUGE-L | 0,45 | 0,447 | 0,458 | 0,453 | 0,339 | 0,411 | 0,399 |
| BERT-F1 | 0,892 | 0,892 | 0,892 | 0,89 | 0,845 | 0,875 | 0,868 |

Based on the ROUGE scores provided in the table, the Graph-Based Document Summarisation using Cross-Encoder and PageRank (GTr-PRSum) achieved the best overall performance.

It obtained the highest ROUGE-1 score (0.658). This means that the graph built using the cross-encoder is particularly good at identifying unigrams (single words) from the reference summaries. Moreover, it also achieved the highest ROUGE-L score (0.456), which shows that the model could retain the structure and coherence of the original text.

Among the graph-based approaches, GTr-PRSum performs slightly better than the other approaches, such as Graph-based summarisation using cosine similarity and HITS algorithm (GCos-HSum). Transformer-based methods achieved BERT-F1 scores (0.892), which were consistent across all of them, indicating their strength in semantic representation.

On the other hand, Naive Bayes model worked moderately well with ROUGE-1 score of 0.592 and ROUGE-L score of 0.411, indicating that it could identify unigrams and some level of structure. Nevertheless, the performance of the SVM model was the weakest overall, as measured by significantly lower scores (e.g., ROUGE-1: 0.507, ROUGE-L: 0.339) than the graph-based methods.

Supervised learning models (SVM and Naive Bayes) produced noticeably bad ROUGE-2 scores, which indicates the difficulty in capturing bigrams (pairs of consecutive words) and local context. On the other hand, the graph-based methods had better ROUGE-2 scores, especially the transformer-based ones.

Overall, the graph built using cross-encoder and PageRank outperformed other methods on unigram matching, coherence, and structural preservation. Results show the efficiency of using graph structures and sophisticated embeddings in solving extractive document summarisation tasks.

### 5.3.2 CNN/DailyMail dataset

Table Scores for CNN/DailyMail dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | GCos-PRSum | GCos\_HSum | GTr\_PRSum | G-Tr\_Hsum | SVMSum | NBSum | KNNSum |
| ROUGE-1 | 0,299 | 0,296 | 0,302 | 0,302 | 0,268 | 0,334 | 0,315 |
| ROUGE-2 | 0,101 | 0,1 | 0,105 | 0,104 | 0,088 | 0,132 | 0,119 |
| ROUGE-L | 0,196 | 0,196 | 0,198 | 0,197 | 0,171 | 0,219 | 0,207 |
| BERT-F1 | 0,782 | 0,782 | 0,783 | 0,782 | 0,766 | 0,79 | 0,783 |

ROUGE scores in the provided table indicate that graph-based extractive summarisation methods achieve strong performance when processing the CNN/DailyMail dataset. The supervised model NBSum delivered the best ROUGE-1 score (0.334) because it proved effective at finding unigrams (individual words) present in reference summaries. Naive Bayes achieved the leading ROUGE-L score of 0.219 during graph-based processing because it maintained the most original text coherence and structure.

Both GCos-PRSum and GTr\_PRSum showed significant performance through their 0.299 and 0.302 ROUGE-1 scores. The embedding techniques combined with graph structures enhance summary quality for these approaches. The BERT-F1 scores (0.783) presented similar results among most graph-based methods because these approaches demonstrate outstanding capabilities in semantic representation and textual meaning understanding.

The SVM model produced lower results, scoring 0.268 using ROUGE-1 and 0.171 using ROUGE-L. SVMSum captures unigrams together with basic structural components, it shows lower performance than graph-based methods. Supervised learning models SVMSum and Naive Bayes, exhibited reduced ROUGE-2 performance because they failed to direct algorithmically pairs of consecutive words correctly.

The graph-based methods that employed transformer-based similarity and PageRank proved most effective for unigram matching and coherence while maintaining structural integrity. Research outcomes demonstrate that document summarisation benefits from graph structures together with advanced embedding techniques, which confirms their position as a future research field.

### 5.3.3 Conclusion

Distinct data properties between the CNN/DailyMail and BBC datasets impact the performance levels when summarising texts. Models easily track and extract important information from shorter BBC dataset texts that use factual writing at a more concise format. The short BBC articles maintain a factual writing style which allows models to identify and extract crucial content. These differences highlight the importance of adjusting summarisation approaches to the specific nature of the text, as models that excel on one dataset may not perform as well on another.

## 5.4 Example Output

Below are example summaries showcasing the quality differences across methods:

### 5.4.1 Original Article:

UK economy facing 'major risks'

The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said.

The group's quarterly survey of companies found exports had picked up in the last three months of 2004 to their best levels in eight years. The rise came despite exchange rates being cited as a major concern. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. It recently forecast economic growth will slow from more than 3% in 2004 to a little below 2.5% in both 2005 and 2006.

Manufacturers' domestic sales growth fell back slightly in the quarter, the survey of 5,196 firms found. Employment in manufacturing also fell and job expectations were at their lowest level for a year.

"Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. "These results reinforce our concern over the sector's persistent inability to sustain recovery." The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted.

The BCC found confidence increased in the quarter across both the manufacturing and service sectors although overall it failed to reach the levels at the start of 2004. The reduced threat of interest rate increases had contributed to improved confidence, it said. The Bank of England raised interest rates five times between November 2003 and August last year. But rates have been kept on hold since then amid signs of falling consumer confidence and a slowdown in output. "The pressure on costs and margins, the relentless increase in regulations, and the threat of higher taxes remain serious problems," BCC director general David Frost said. "While consumer spending is set to decelerate significantly over the next 12-18 months, it is unlikely that investment and exports will rise sufficiently strongly to pick up the slack."

### 5.4.2 Reference summary:

"Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said.The BCC found confidence increased in the quarter across both the manufacturing and service sectors although overall it failed to reach the levels at the start of 2004.The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted.The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said.However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow.The reduced threat of interest rate increases had contributed to improved confidence, it said.The rise came despite exchange rates being cited as a major concern.

### 5.4.3 GRAPH\_COSINE\_PAGERANK SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. "Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted. "While consumer spending is set to decelerate significantly over the next 12-18 months, it is unlikely that investment and exports will rise sufficiently strongly to pick up the slack."

### 5.4.4 GRAPH\_COSINE\_HITS SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. "Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted. "While consumer spending is set to decelerate significantly over the next 12-18 months, it is unlikely that investment and exports will rise sufficiently strongly to pick up the slack."

### 5.4.5 GRAPH\_TRANSFORMER\_PAGERANK SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. The rise came despite exchange rates being cited as a major concern. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. "Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted.

### 5.4.6 GRAPH\_TRANSFORMER\_HITS SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. The rise came despite exchange rates being cited as a major concern. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. "Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted.

### 5.4.7 SVM SUMMARY

Manufacturers' domestic sales growth fell back slightly in the quarter, the survey of 5,196 firms found. Employment in manufacturing also fell and job expectations were at their lowest level for a year. "Despite some positive news for the export sector, there are worrying signs for manufacturing," the BCC said. "These results reinforce our concern over the sector's persistent inability to sustain recovery." The outlook for the service sector was "uncertain" despite an increase in exports and orders over the quarter, the BCC noted.

### 5.4.8 NAIVE\_BAYES SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. The group's quarterly survey of companies found exports had picked up in the last three months of 2004 to their best levels in eight years. The rise came despite exchange rates being cited as a major concern. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. It recently forecast economic growth will slow from more than 3% in 2004 to a little below 2.5% in both 2005 and 2006.

### 5.4.9 KNN SUMMARY

UK economy facing 'major risks' The UK manufacturing sector will continue to face "serious challenges" over the next two years, the British Chamber of Commerce (BCC) has said. However, the BCC found the whole UK economy still faced "major risks" and warned that growth is set to slow. It recently forecast economic growth will slow from more than 3% in 2004 to a little below 2.5% in both 2005 and 2006. Manufacturers' domestic sales growth fell back slightly in the quarter, the survey of 5,196 firms found. The Bank of England raised interest rates five times between November 2003 and August last year.

### 5.4.10 Conclusion

Graph-Based Extractive Summarisation using Transformer embeddings in combination with PageRank-based ranking achieves superior results than other measured approaches in text summarisation. During summarisation, Graph-Based Extractive Summarisation outperformed both Naïve Bayes and SVM and k-Nearest Neighbours, because of its superior retention of logical structure. Advanced embeddings that work with graph structures enable better summary generations. Advanced Graph structure models, together with embedding systems, show evidence of improving text summarisation system performance standards.

## 5.5 Conclusion

Using Transformer embeddings allows Graph-Based Extractive Document Summarisation to successfully produce meaningful, concise summaries. The proposed method achieved superior results compared to Naïve Bayes and SVM as well as k-Nearest Neighbours models, specifically in unigram matching, document preservation, and semantic relationship building, due to its capability for analysis. For both CNN/DailyMail and BBC datasets, the graph-based algorithm showed a stable performance following the usage of advanced Transformer embeddings and the evaluation metrics ROUGE and BERTScore. The CNN/DailyMail dataset generated slightly inferior ROUGE results to those of the BBC dataset because its long and complex articles posed challenges for evaluation. The findings form a strong foundation for further exploration, while the limitations and potential future directions will be addressed in the subsequent sections to provide a more comprehensive understanding of the proposed method's performance and scope.

# Discussion

## 6.1 Introduction

This section explores the implications of the findings from the analysis and evaluation section of the Graph-Based Extractive Document Summarisation model using Transformer embeddings. The discussion highlights the limitations of the method and potential solutions, along with considerations for scalability and adaptability. These elements provide a comprehensive understanding of the areas that require further development.

## 6.2 Limitations

The model appears to perform less efficiently for preserving flow and critical information extraction in articles with extended lengths. The CNN/DailyMail dataset proved to be problematic for summary models because it presented longer and denser articles, which led to lower ROUGE scores when compared against the BBC dataset.

Summary accuracy is negatively impacted when articles contain inconsistent content or poor writing quality. The computational requirements of using transformer embeddings are high. The accuracy of summaries suffers when articles contain inconsistent content or show poor writing quality. The capability of the model to generate unnecessary repetition exists as a significant drawback when producing summaries.

The removal process for redundant sentences may not prevent complete repetition within summaries when articles consist of extended sections containing duplicated information across various sentences. The quality and density of summaries may suffer as a result. The assessment capabilities of the method depend on the reliability of the received data. Summary information tends to become less accurate when reports present inconsistent writing or demonstrate poor quality writing. Quantitative metrics assess summary quality, but they fall short when measuring against human assessment standards.

The metrics used for evaluation ROUGE and BERTScore exhibit specific constraints during evaluation processes. The evaluation metrics create measurable scores which represent summary quality but fail to entirely reflect human evaluators' judgments.

The graph-based approach utilising transformer embeddings shows promising results yet demonstrates inconsistent effectiveness across diverse textual genres such as scientific papers and legal documents, and social media content.

Another limitation appears because the method struggles to adapt itself successfully to different domains while handling varying text types. The graph-based approach using transformer embeddings reveals potential, yet its performance can change depending on text genres such as scientific papers, social media content or legal documents. Method adaptation across different domains needs thorough testing before achieving uniform results.

## 6.2.1 Potential Solutions

Several prospective solutions exist to fix the limitations discovered throughout the evaluation of Graph-Based Extractive Document Summarisation. Using better sentence splitting methods alongside hierarchical summarisation algorithms enables users to handle complex articles while sustaining better article coherence in the final results. Model optimisation methods, such as pruning along with distillation, allow transformer embedding optimisations for real-time accessible applications. Advanced redundancy algorithms and semantic grouping systems minimise duplicated content in summary text. The handle of diverse textual content improves when combining sturdy data cleansing methods with model optimisation for particular domains. During evaluation, historical human assessment provides reviewers with tools to measure summary performance through rating organisation coherence and information density and clarity. Real-time operations using extensive datasets become challenging for the graph-based approach since it needs large computational resources. Exploring optimised solutions for the Graph-Based Extractive Document Summarisation method will improve its robust operationality when working with different summarisation tasks.

## 6.3 Scalability and Adaptability

Processing massive or streaming document datasets becomes difficult because of the computational requirements associated with the graph-based method. The flexibility across domains in unsupervised approaches diminishes when they are used in specialist or informal settings. This domain-independent unsupervised operation leads to imprecise results in specific text-style contexts with complex structures despite its advantages in domain openness. The need arises for finding an equilibrium point between scalability and domain-specific adaptability which produces dependable outcomes in different datasets.

## 6.3.1 Potential Solutions

The evaluation of the Graph-Based Extractive Document Summarisation method reveals multiple limitations which can be remedied through different proposed solutions. Better hierarchical summarisation methods combined with improved sentence splitting technologies enable unified large text document processing and analysis. Real-time systems benefit from better accessibility through distillation combined with model pruning techniques, which reduce the requirements for transformers.

Improved semantic clustering algorithms should be deployed to remove repetitive content in summary outputs. Pre-processing methods trained for specific text types and data cleaning protocols enhance our ability to process texts. Evaluation protocols assist systems to achieve better outputs by evaluating both clear content quality and informative content quality together with coherence assessments. Better scalability results from domain testing combined with optimised parameter settings across different datasets. Studies of solution possibilities enhance the stability of Graph-Based Extractive Document Summarisation while enabling wider applications in different industries.

## Conclusion

This research investigates key limitations of Graph-Based Extractive Summarisation as well as its potential areas of advancement through transformer-embedded techniques. Developed solutions represent new routes to address preventative constraints so new summarisation approaches can scale across multiple fields and contexts. Future methods aim to develop summary processing technologies that deliver the best functionality within multiple application scenarios.

# Conclusions and Recommendations

## 7.1 Conclusions

A study explores how to develop and test document summarisation model based on graph structures, along with transformer embedding technology. The research targeted higher accuracy and processing speed for generalisation assignments through deep learning contextual embeddings combined with graph-based ranking algorithms. To obtain complete evaluation benchmarks the proposed method underwent testing with both CNN/DailyMail and BBC datasets.

Generalisations obtained through the combination of the graph method with extended transformer embeddings proved effective at producing coherent and compact results. The proposed method excelled at unigram matching and structural preservation while achieving better semantic similarity than traditional methods based on Naive Bayes, KNN and SVM. The CNN/DailyMail dataset produced varied performance results since its articles were longer as well as more complex when compared to the other datasets.

The positive results from the methodology revealed two main limitations, such as the high computational intensity and the need for repeated domain-focused generalisations. Research progress in document processing depends on the integration of optimised embedding functions from human feedback systems with preprocessing developments.

Overall, the research contributes to the field of automatic text summarisation by presenting an advanced, hybrid approach that balances linguistic coherence, computational efficiency, and semantic depth. The research results create a strong basis for both new investigations and the creation of better summarisation techniques.

## 7.2 Recommendations

The study findings and its acknowledged constraints will support the production of practical implementation procedures along with theoretical recommendations.

Results from human evaluation combined with automatic metrics ROUGE and BERTScore should be used to evaluate the performance of future work, since human evaluators can better understand text quality by comparing text fluency and coherence versus human reference documents.

The model reveals relevant capabilities for assessing legal documents along with scientific research documents and medical documentation, to assist with usability tests and optimisation decisions.

Transformer models offer high complexity, enabling future optimisation through techniques including pruning and quantisation and knowledge distillation, which will enable real-time summarisation on low-resource systems.

The addition of transformer-based decoders for abstractive summarisation would likely lead to more natural and expressive summaries than extractive approaches alone.

The combination of model optimisation techniques through pruning and quantisation alongside knowledge distillation will speed up low-resource platforms when driven by transformer models.

Finally, user-defined parameters like summary length, content focus or readability level would enable more interactivity and adaptability of the model to specific user needs and thereby increase its practicality on different applications. The directions provided in these directions provide useful paths for improving and expanding the capabilities of hybrid summarisation models.

# Appendix A

import os

import re

import time

import torch

import numpy as np

import networkx as nx

from datetime import datetime

from datasets import load\_dataset

from functools import lru\_cache, partial

from concurrent.futures import ThreadPoolExecutor

from sklearn.metrics.pairwise import cosine\_similarity

from nltk.tokenize import sent\_tokenize

from rouge\_score import rouge\_scorer

from sentence\_transformers import SentenceTransformer, CrossEncoder

from bert\_score import BERTScorer

from tqdm.auto import tqdm

from collections import defaultdict

import warnings

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive\_bayes import MultinomialNB

from sklearn.preprocessing import MinMaxScaler

from huggingface\_hub import login

# Configuration

os.environ["TOKENIZERS\_PARALLELISM"] = "false"

os.environ["TOKENIZERS\_NO\_THREADING"] = "1"

torch.set\_grad\_enabled(False)

torch.set\_float32\_matmul\_precision('medium')

warnings.filterwarnings("ignore")

# Initialize models

DEVICE = 'cuda' if torch.cuda.is\_available() else 'cpu'

SBERT\_MODEL = 'all-MiniLM-L6-v2'

CROSS\_ENCODER\_MODEL = 'cross-encoder/stsb-distilroberta-base'

BERT\_SCORER\_MODEL = 'distilbert-base-uncased'

print(f"Initializing models on {DEVICE}...")

try:

sbert\_model = SentenceTransformer(SBERT\_MODEL, device=DEVICE)

cross\_encoder = CrossEncoder(CROSS\_ENCODER\_MODEL, device='cpu')

bert\_scorer = BERTScorer(lang="en", model\_type=BERT\_SCORER\_MODEL, device=DEVICE)

except Exception as e:

print(f"Error loading models: {str(e)}")

print("Using fallback models...")

sbert\_model = SentenceTransformer('paraphrase-MiniLM-L6-v2', device=DEVICE)

cross\_encoder = CrossEncoder('cross-encoder/ms-marco-MiniLM-L-6-v2', device='cpu')

bert\_scorer = BERTScorer(lang="en", model\_type='microsoft/deberta-base-mnli', device=DEVICE)

# Data loading functions

def load\_bbc\_dataset(path="BBC\_News\_Summary\_01", max\_articles=500):

"""Load BBC News dataset from local files"""

categories = ["business", "entertainment", "politics", "sport", "tech"]

data = []

for category in categories:

articles\_path = os.path.join(path, "News Articles", category)

summaries\_path = os.path.join(path, "Summaries", category)

if not os.path.exists(articles\_path):

continue

for file\_name in os.listdir(articles\_path):

if len(data) >= max\_articles:

break

if file\_name.endswith(".txt"):

article\_path = os.path.join(articles\_path, file\_name)

summary\_path = os.path.join(summaries\_path, file\_name)

try:

with open(article\_path, "r", encoding='utf-8') as f:

article\_text = f.read().strip()

with open(summary\_path, "r", encoding='utf-8') as f:

summary\_text = f.read().strip()

data.append({

"category": category,

"article": article\_text,

"summary": summary\_text

})

except Exception as e:

print(f"Error loading {file\_name}: {str(e)}")

continue

return data

def load\_cnn\_dailymail\_dataset(split='train[:3]'):

"""Load CNN/DailyMail dataset from HuggingFace"""

try:

dataset = load\_dataset('cnn\_dailymail', '3.0.0', split=split)

# Standardize the field names

processed\_data = []

for item in dataset:

processed\_data.append({

"article": item["article"],

"summary": item["highlights"]

})

return processed\_data

except Exception as e:

print(f"Error loading CNN/DailyMail dataset: {str(e)}")

return []

# Preprocessing

@lru\_cache(maxsize=5000)

def cached\_encode(sentence):

return sbert\_model.encode(sentence, convert\_to\_tensor=False)

def preprocess\_article(article, max\_length=2000):

article = re.sub(r'\s+', ' ', article)[:max\_length]

return sent\_tokenize(article)

# Graph-based methods

def calculate\_similarity\_matrix(sentences, method='cosine'):

if method == 'cosine':

embeddings = np.array([cached\_encode(s) for s in sentences])

return cosine\_similarity(embeddings)

elif method == 'transformer':

pairs = [(i, j) for i in range(len(sentences)) for j in range(i+1, len(sentences))]

batch\_size = min(32, len(pairs))

with ThreadPoolExecutor() as executor:

process\_fn = partial(process\_pair, sentences=sentences)

similarity\_scores = list(executor.map(process\_fn, pairs, chunksize=batch\_size))

similarity\_matrix = np.zeros((len(sentences), len(sentences)))

for (i, j), score in zip(pairs, similarity\_scores):

similarity\_matrix[i][j] = score

similarity\_matrix[j][i] = score

return similarity\_matrix

def process\_pair(pair, sentences):

i, j = pair

return cross\_encoder.predict([(sentences[i], sentences[j])], show\_progress\_bar=False)[0]

def build\_and\_rank\_graph(similarity\_matrix, method='pagerank'):

threshold = np.percentile(similarity\_matrix, 80)

similarity\_matrix[similarity\_matrix < threshold] = 0

np.fill\_diagonal(similarity\_matrix, 0)

graph = nx.from\_numpy\_array(similarity\_matrix)

if method == 'pagerank':

scores = nx.pagerank(graph, weight='weight')

elif method == 'hits':

\_, scores = nx.hits(graph)

else:

raise ValueError(f"Unknown method: {method}")

return scores

def remove\_redundant\_sentences(sentences, embeddings=None, scores=None, threshold=0.7, summary\_size=5):

"""Flexible redundancy removal with summary size limit"""

selected = []

# Create list of (score, index, sentence) tuples

items = [

(scores[i] if scores is not None else 1, i, sent)

for i, sent in enumerate(sentences)

]

# Sort by score if available

items.sort(reverse=True, key=lambda x: x[0])

for score, idx, sent in items:

if len(selected) >= summary\_size:

break

if embeddings is not None:

# Fast path with pre-computed embeddings

is\_redundant = any(

cosine\_similarity([embeddings[idx]], [embeddings[s[0]]])[0][0] >= threshold

for s in selected

)

else:

# Slow path with on-demand encoding

is\_redundant = any(

cosine\_similarity([sbert\_model.encode(sent)],

[sbert\_model.encode(selected\_sent)])[0][0] >= threshold

for \_, selected\_sent in selected

)

if not is\_redundant:

selected.append((idx, sent))

# Return in original order

return [sent for idx, sent in sorted(selected, key=lambda x: x[0])]

# Classifier-based methods

def summarize\_with\_classifier(sentences, embeddings, classifier\_type='svm', summary\_size=5):

# Prepare features and pseudo-labels

features = embeddings

labels = np.zeros(len(sentences))

labels[:summary\_size] = 1 # First N sentences as positive examples

# Train classifier

if classifier\_type == 'svm':

model = SVC(kernel='linear', probability=True)

elif classifier\_type == 'naive\_bayes':

scaler = MinMaxScaler()

features = scaler.fit\_transform(features) # NB requires non-negative features

model = MultinomialNB()

elif classifier\_type == "knn":

n\_neighbors = min(9, len(sentences) - 1)

model = KNeighborsClassifier(n\_neighbors=n\_neighbors)

else:

raise ValueError(f"Unknown classifier: {classifier\_type}")

model.fit(features, labels)

# Predict probabilities

probabilities = model.predict\_proba(features)[:, 1]

selected\_sentences = remove\_redundant\_sentences(

sentences=sentences,

embeddings=embeddings,

scores=probabilities, # Classifier scores

threshold=0.7, # Slightly lower threshold for ML

summary\_size=summary\_size

)

return ' '.join(selected\_sentences)

# Unified summarization function

def summarize\_article(article, method='graph', similarity\_method='cosine',

ranking\_method='pagerank', summary\_size=5):

sentences = preprocess\_article(article)

if len(sentences) < 3:

return ""

embeddings = np.array([cached\_encode(s) for s in sentences])

if method == 'graph':

similarity\_matrix = calculate\_similarity\_matrix(sentences, method=similarity\_method)

scores = build\_and\_rank\_graph(similarity\_matrix, method=ranking\_method)

selected\_sentences = remove\_redundant\_sentences(

sentences=sentences,

embeddings=embeddings,

scores=scores, # Graph-based scores

threshold=0.7,

summary\_size=summary\_size

)

return ' '.join(selected\_sentences)

elif method in ['svm', 'naive\_bayes', 'knn']:

return summarize\_with\_classifier(sentences, embeddings,

classifier\_type=method,

summary\_size=summary\_size)

else:

raise ValueError(f"Unknown method: {method}")

# Evaluation

def evaluate\_batch(dataset, methods\_config, dataset\_name="Unknown", summary\_size=5):

results = defaultdict(lambda: defaultdict(list))

scorer = rouge\_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL', 'rougeLsum'], use\_stemmer=True)

for sample in tqdm(dataset, desc="Processing articles"):

article = sample['article']

reference = sample['summary']

for config in methods\_config:

method\_type = config['method']

try:

summary = summarize\_article(

article,

method=method\_type,

similarity\_method=config.get('similarity', 'cosine'),

ranking\_method=config.get('ranking', 'pagerank'),

summary\_size=summary\_size

)

# ROUGE scores

rouge = scorer.score(reference.lower(), summary.lower())

for key in ['rouge1', 'rouge2', 'rougeL', 'rougeLsum']:

results[str(config)][f'rouge\_{key}'].append(rouge[key].fmeasure)

# BERTScore

P, R, F1 = bert\_scorer.score([summary], [reference])

results[str(config)]['bert\_p'].append(P.mean().item())

results[str(config)]['bert\_r'].append(R.mean().item())

results[str(config)]['bert\_f1'].append(F1.mean().item())

except Exception as e:

print(f"Error with {config}: {str(e)}")

continue

return results

def save\_results(results, dataset\_name, dataset\_len, output\_dir="results"):

"""Save evaluation results to file"""

# Calculate averages

final\_results = {}

for config, metrics in results.items():

final\_results[config] = {

'rouge1': np.mean(metrics['rouge\_rouge1']),

'rouge2': np.mean(metrics['rouge\_rouge2']),

'rougeL': np.mean(metrics['rouge\_rougeL']),

'rougeLsum': np.mean(metrics['rouge\_rougeLsum']),

'bert\_f1': np.mean(metrics['bert\_f1'])

}

# Print and save results

print("\n=== Final Results ===")

timestamp = datetime.now().strftime("%Y%m%d\_%H%M%S")

if not os.path.exists(output\_dir):

os.makedirs(output\_dir)

filename = os.path.join(output\_dir, f"summary\_results\_{timestamp}.txt")

with open(filename, "w") as f:

f.write(f"Dataset: {dataset\_name}\n")

f.write(f"Number of articles: {dataset\_len}\n")

f.write("\n=== Evaluation Results ===\n\n")

for config, scores in final\_results.items():

line = (f"{config:<72}: "

f"ROUGE-1 = {scores['rouge1']:.3f} | "

f"ROUGE-2 = {scores['rouge2']:.3f} | "

f"ROUGE-L = {scores['rougeL']:.3f} | "

f"ROUGE-Lsum = {scores['rougeLsum']:.3f} | "

f"BERT-F1 = {scores['bert\_f1']:.3f}")

print(line)

f.write(line + "\n")

return filename

if \_\_name\_\_ == "\_\_main\_\_":

start\_time = time.time()

# Configuration

DATASET\_CHOICE = "bbc" # "bbc" or "cnn"

DATASET\_SIZE = 5000

SUMMARY\_SIZE = 5 # 5 for bbc and 3 for cnn

# Load data

print("Loading dataset...")

try:

if DATASET\_CHOICE == "bbc":

data = load\_bbc\_dataset(max\_articles=500)

dataset = data[:DATASET\_SIZE]

dataset\_name = "BBC News Summary"

elif DATASET\_CHOICE == "cnn":

dataset = load\_cnn\_dailymail\_dataset(split=f'train[:{DATASET\_SIZE}]')

dataset\_name = "CNN/DailyMail"

else:

raise ValueError("Invalid dataset choice. Use 'bbc' or 'cnn'")

if not dataset:

raise ValueError("No data loaded")

except Exception as e:

print(f"Error loading dataset: {str(e)}")

exit(1)

# Define evaluation configurations

methods\_config = [

# Graph-based methods

{'method': 'graph', 'similarity': 'cosine', 'ranking': 'pagerank'},

{'method': 'graph', 'similarity': 'cosine', 'ranking': 'hits'},

{'method': 'graph', 'similarity': 'transformer', 'ranking': 'pagerank'},

{'method': 'graph', 'similarity': 'transformer', 'ranking': 'hits'},

# Classifier methods

{'method': 'svm'},

{'method': 'naive\_bayes'},

{'method': 'knn'},

]

# Run evaluation

try:

results = evaluate\_batch(dataset, methods\_config, dataset\_name, SUMMARY\_SIZE)

results\_file = save\_results(results, dataset\_name, len(dataset))

print(f"\nResults saved to: {results\_file}")

print(f"Total execution time: {time.time() - start\_time:.2f} seconds")

except Exception as e:

print(f"Error during evaluation: {str(e)}")

# Bibliography

|  |  |
| --- | --- |
| [1] | C. D. A. D. M. D. A. a. S. A. Mallick, “Graph-based text summarization using modified TextRank,” *Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018 ,* pp. 137-146, 2019. |
| [2] | G. a. R. D. Erkan, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of artificial intelligence research,* vol. 22, pp. 457-479, 2024. |
| [3] | J. Devlin, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. |
| [4] | N. a. G. I. Reimers, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2019. |
| [5] | L. Page, “The PageRank citation ranking: Bringing order to the web,” Stanford InfoLab, 1999. |
| [6] | M. Prajapati, “A survey paper on hyperlink-induced topic search (HITS) algorithms for web mining,” *Int J Eng,* vol. 1, no. 2, p. 8, 2012. |
| [7] | Python, “Python,” [Online]. Available: https://www.python.org/. |
| [8] | P. Sharif, “BBC News Summary,” [Online]. Available: https://www.kaggle.com/datasets/pariza/bbc-news-summary. [Accessed November 2024]. |
| [9] | K. M. K. T. G. E. E. L. K. W. S. M. B. P. Hermann, “Teaching Machines to Read and Comprehend,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1693--1701. |
| [10] | A. Karotia and S. Susan, “CovSumm: An Unsupervised Transformer-Cum-Graph-Based Hybrid Document Summarization Model for CORD-19,” *The Journal of Supercomputing,* vol. 79, no. 14, pp. 16328-16350, 2023. |
| [11] | T. K. V. W. F. W. K. Q. &. A. Y. Zhang, “Bertscore: Evaluating text generation with bert,” *arXiv preprint,* vol. arXiv:1904.09675., 2019. |
| [12] | C. V. V. Cortes, “Support-vector networks.,” *Machine learning,* vol. 20, pp. 273-297, 1995. |
| [13] | A. K. N. McCallum, “A comparison of event models for naive bayes text classification,” *AAAI-98 workshop on learning for text categorization ,* vol. 752, no. 1, pp. 41-48, 1998. |
| [14] | T. P. H. Cover, “Nearest neighbor pattern classification,” *EEE transactions on information theory ,* vol. 13, no. 1, pp. 21-27, 1967. |
| [15] | M. K. G. M. M. Kirmani, “Analysis of Abstractive and Extractive Summarization Methods,” *International Journal of Emerging Technologies in Learning,* vol. 19, no. 1, 2024. |
| [16] | K. &. K. P. &. K. S. &. A. A. Rajendran, “TEXT SUMMARIZATION,” 2023. |
| [17] | E. C. L. M. N. a. F. A. Baralis, “GraphSum: Discovering correlations among multiple terms for graph-based summarization,” *Information Sciences,* vol. 249, pp. 96-109, 2013. |
| [18] | J. B. S. B. M. B. P. B. A. C. S. W. J. a. M. A. Verma, “Graph-Based Extractive Text Summarization Sentence Scoring Scheme for Big Data Applications.,” *Information,* vol. 14, no. 9, p. 472, 2023. |
| [19] | A. Abdi, S. Hashemi and M. A. Nematbakhsh, “ASMUS: Automatic sentiment-oriented summarization of multi-documents using soft computing,” *Soft Computing,* vol. 23, no. 11, p. 4051–4067, 2019. |
| [20] | L. C. P. a. S. T. Muller, “Transformers and cortical waves: encoders for pulling in context across time,” *Trends in neurosciences,* 2024. |
| [21] | Y. a. M. L. Liu, “Text summarization with pretrained encoders,” 2019. |
| [22] | L. Bacco, A. Cimino, F. Dell’Orletta and M. Merone, “Explainable sentiment analysis: a hierarchical transformer-based extractive summarization approach,” *Electronics,* vol. 10, no. 18, p. 2195, 2021. |
| [23] | J. Kupiec, J. Pedersen and F. Chen, “A trainable document summarizer,” in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, WA, 1995. |
| [24] | X. L. Z. W. T. L. S. Mao, “Extractive Summarization Using Supervised and Unsupervised Learning,” *Expert Systems with Applications,* vol. 133, p. 173–181, 2019. |
| [25] | P. Gowrishankar, “Newspaper Text Summarization - CNN/DailyMail,” 2021. [Online]. Available: https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail. [Accessed February 2025]. |
| [26] | D. a. C. P. Greene, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006. |
| [27] | JetBrains, “PyCharm,” 2019. [Online]. Available: https://www.jetbrains.com/pycharm/. |
| [28] | A. Al-Taani, “Automatic Text Summarization Approaches,” *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS),* p. 93–94, 2017. |
| [29] | C. S. S. U. K. I. H. Nobata, “A Summarization System with Categorization of Document Sets,” *NTCIR,* NTCIR. |
| [30] | N. J. A. Bhatia, “Trends in Extractive and Abstractive Techniques in Text Summarization,” *International Journal of Computer Applications,* vol. 117, no. 6, 2015. |
| [31] | A. B. F. a. D. B. Bougouin, “Topicrank: Graph-based topic ranking for keyphrase extraction.,” *nternational joint conference on natural language processing ,* pp. 543-551, 2013. |
| [32] | A. N. E. S. A. &. A. Widyassari, “The 7-Phases Preprocessing Based On Extractive Text Summarization,” *Seventh International Conference on Informatics and Computing (ICIC),* pp. 1-8, 2022. |
| [33] | Y. Ledeneva, “Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences,” *Mexican International Conference on Artificial Intelligence,* 2008. |
| [34] | C. N. S. A. R. K. L. S. N. M. M. Y. Z. W. L. a. P. J. L. Raffel, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research,* vol. 21, no. 140, pp. 1-67, 2020. |
| [35] | G. a. R. M. a. S. G. C. a. S. Y. a. S. G. a. M. P. K. R. a. R. G. D. a. J. R. H. a. P. B. a. W. W. a. V. A. V. a. G. Yenduri, “GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions,” *IEEE Access,* vol. 12, pp. 54608-54649, 2024. |
| [36] | N. S. A. R. K. L. S. N. M. M. Y. Z. W. L. P. J. L. Colin Raffel, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of machine learning research,* vol. 21, no. 140, pp. 1-67, 2020. |
| [37] | A. Awan, “Naive Bayes Classifier Tutorial: with Python Scikit-learn,” 3 March 2023. [Online]. Available: https://www.datacamp.com/tutorial/naive-bayes-scikit-learn. [Accessed 1 January 2025]. |
| [38] | M. S. a. Y.-K. N. Pera, “A Naive Bayes classifier for web document summaries created by using word similarity and significant factors,” *International Journal on Artificial Intelligence Tools,* vol. 19, no. 04, pp. 465-86, 2010. |
| [39] | Y. C. a. S. A. H. a. S. R. Joty, “A SVM-Based Ensemble Approach to Multi-Document Summarization,” *Advances in Artificial Intelligence, Canadian AI 2009, Lecture Notes in Computer Science,* vol. 5549, 2009. |
| [40] | Y. L. Y. Y. H. X. W. L. Z. L. X. C. Yukai Yao, “K-SVM: An Effective SVM Algorithm Based on K-means Clustering,” *Journal of computers,* vol. 8, no. 10, 2013. |