AUTOMATIC GENERATION OF RESEARCH PAPER ABSTRACTS USING DEEP-HYBRID MODELS

R.P.D. Kumarasinghe

219354V

Master of Science in Computer Science

Department of Computer Science & Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

February 2025

AUTOMATIC GENERATION OF RESEARCH PAPER ABSTRACTS USING DEEP-HYBRID MODELS

R.P.D. Kumarasinghe

219354V

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science & Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

February 2025

DECLARATION

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:  Date: 28/02/2025

The supervisor should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science in Computer Science Thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Nisansa de Silva

Signature of the Supervisor: Date:

DEDICATION

I dedicate this to all who strive to seek knowledge, believing that every step forward is a step towards a better world.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my research supervisor, Dr. Nisansa de Silva, for their unwavering guidance, support, and encouragement throughout the course of this thesis. Their insightful advice, constructive feedback, and immense patience have been invaluable in shaping the direction and quality of this research. I am deeply thankful for the time and effort they dedicated to helping me grow academically and professionally, and for always challenging me to think critically and aim for excellence.

Lastly, I extend my appreciation to my family and friends for their constant encouragement and understanding during this challenging process. Their support made this accomplishment possible.

ABSTRACT

Condensing important information into a summary is crucial for readers navigating lengthy documents. In the context of research papers, the **abstract** serves as a concise overview of the study. This thesis focuses on enhancing research paper summarization by introducing a novel section-wise relevance matrix. To address the token size limitations of Large Language Models (LLMs) , such as GPT-Neo, we developed a two-fold approach. First, we employed extractive summarization to condense lengthy texts into key sentences, followed by the application of abstractive summarization to generate coherent and concise summaries from these extracts.

Our approach, combining both extractive and abstractive techniques, leverages section-wise involvement ratios, with particular attention to the abstract section, improving the accuracy and quality of generated summaries. We introduced a pioneering dataset of research papers organized into sections, which plays a crucial role in this summarization process. Experimental results demonstrated that our method produces high-quality summaries while effectively overcoming token limitations, offering significant potential for summarizing long documents in low-resource and cost-effective environments.

However, challenges arise when section-wise segmentation is unclear, impacting the accuracy of summaries. This research underscores the need for further refinements and offers a promising framework for enhancing summarization techniques, benefiting researchers, educators, and information seekers alike.

Keywords: NLP, ATS, Summarization, Text Generation, LLM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
BLEU	Bilingual Evaluation Understudy
CPUs	Central Processing Units
GPUs	Graphics Processing Units
I/O	Input/Output
LCS	Longest Common Subsequence
LLMs	Large Language Models
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
TPUs	Tensor Processing Units

CHAPTER 1

INTRODUCTION

1.1 Background

The *abstract* of a research paper serves as a crucial component, offering a succinct summary that encapsulates the paper's key elements namely, the problem it addresses, the proposed solution, and the results achieved. By design, the abstract is meant to be both concise and informative, enabling readers to quickly grasp the essence of the research without delving into the entire paper. Given its importance, the abstract often dictates whether a reader will invest time in reading the full document, making it a vital part of academic writing.

Text summarization, a prominent domain within Natural Language Processing (NLP) , is highly relevant in contexts where large volumes of text need to be condensed into shorter, yet meaningful, summaries. The ability to efficiently summarize text is not only valuable in academic settings but also in industries like journalism, law, and business, where quick access to key information is essential. Within the realm of text summarization, two primary techniques have emerged: *extraction* and *abstraction* summarization.

Extractive summarization focuses on selecting and concatenating pre-existing sentences, phrases, or words directly from the source text to form a coherent summary. This approach is relatively straightforward as it relies on the text's original structure, ensuring that the summary maintains the context and meaning of the source material. However, while extractive summaries preserve accuracy, they may lack fluidity and coherence, as the selected pieces may not always align perfectly when combined.

Abstractive summarization, in contrast, aims to generate new sentences that encapsulate the core ideas of the source text. This technique goes beyond mere extraction, employing advanced algorithms to create a semantic representation of the content. Once this representation is established, natural language generation techniques are used to produce a summary that is not only accurate but also more fluid and concise than its extractive counterpart. Although abstractive summarization can yield more readable and coherent summaries, it is inherently more complex and computationally intensive, requiring sophisticated models that understand and generate human language effectively.

This research presents a novel approach that integrates both extractive and abstractive techniques into a hybrid mechanism designed to generate the abstract sections of scientific papers. By combining the strengths of both methods, the hybrid approach aims to produce summaries that are both coherent and contextually accurate, addressing the limitations associated with using either method in isolation. To evaluate the ef-

fectiveness of this hybrid approach, the research introduces a newly developed dataset specifically tailored to the task of abstract generation in scientific literature. This also explores various strategies for optimizing the hybrid mechanism, offering insights into the potential benefits and challenges of integrating extractive and abstractive summarization techniques in the context of scientific research.

1.2 Research Problem

As discussed in Section 1.1, the *abstract* section of a research paper plays a critical role in conveying a concise and informative summary of the study. It serves as the first point of contact for readers, providing a snapshot of the research problem, methodology, findings, and conclusions. Despite its importance, crafting an effective abstract is often a daunting task for researchers. The process requires not only a deep understanding of the study but also the ability to distill complex information into a brief yet comprehensive narrative. This challenge is further compounded by the need to ensure clarity, coherence, and appeal, all within a limited word count.

Given the high stakes involved in abstract writing, researchers often invest significant time and effort into this process, revisiting and refining their summaries multiple times before finalizing them. This time-consuming task can divert valuable resources from other critical aspects of the research process, such as data analysis, manuscript preparation, and peer review. Moreover, the pressure to produce a well-crafted abstract can be particularly overwhelming for early-career researchers or non-native English speakers, who may struggle with articulating their findings in a concise and compelling manner.

The primary objective of this research is to alleviate the burden on researchers by developing a method for automatically generating abstract sections based on the content of the subsequent sections of the paper. This automated approach is designed to analyze the full text of the paper, extracting key information and synthesizing it into a draft abstract that accurately reflects the core contributions of the study. By leveraging advanced Natural Language Processing (NLP) techniques, this system aims to produce abstracts that are not only informative and coherent but also tailored to the specific structure and style of scientific writing.

The goal of this research is twofold: first, to save researchers time and effort by providing a solid foundation for their abstracts; and second, to enhance the overall quality and consistency of scientific abstracts across disciplines. By allowing researchers to focus on fine-tuning the generated abstracts rather than creating them from scratch, this approach has the potential to streamline the manuscript preparation process and reduce the cognitive load associated with abstract writing. Ultimately, this research seeks to contribute to the broader field of academic writing by offering a tool that supports researchers in effectively communicating their findings to a global audience.

1.3 Research Objectives

The overarching objective of this research is to develop methods for generating *abstract*s for research papers accurately and efficiently, leveraging improved and adapted text summarization models. The specific objectives are as follows:

1. Creating a Sufficient Dataset: Develop a comprehensive dataset tailored for the task of *abstract* generation. This dataset will be section-wise broken down to facilitate detailed analysis and model training.

   This involves collecting and curating research papers with clearly defined sections such as Introduction, Related Work, Methodology, Experiments, Results, and Conclusion.

2. Evaluating Existing Summarization Technologies: Assess the performance of state-of-the-art text summarization technologies on the newly created dataset as well as other comparable datasets.

   This evaluation will include both extractive and abstractive models, analyzing their strengths and weaknesses in generating high-quality abstracts.

3. Developing Automatic Summarization Models: Create advanced summarization models capable of generating *abstract*s automatically.

   These models will integrate both extractive and abstractive techniques, leveraging the section wise importance to ensure summaries are concise, informative, and contextually relevant. The models will be designed to function efficiently even in resource-constrained environments, broadening their applicability.

By achieving these objectives, this research aims to advance the technical capabilities of NLP models in the domain of research paper summarization. The development of a new dataset and the hybrid summarization model will contribute to the field by providing a robust solution for generating high-quality abstracts. Furthermore, making these technologies accessible to low-resource settings will help democratize NLP advancements, bridging the gap between technological benefits and broader inclusivity.

In the following sections, we will delve into the detailed mechanisms of our approach, evaluate its performance against contemporary solutions, and discuss its potential applications in both academic and practical contexts. Through this work, we aim to enhance the efficiency and effectiveness of summarization models while making these technologies more accessible to a wider range of users.

CHAPTER 2

PUBLICATIONS

Candidate's accepted and published papers related to this research are listed below.

- **Dushan Kumarasinghe**, Nisansa de Silva "Automatic Generation of Abstracts for Research Papers" 34th annual Conference on Computational Linguistics and Speech Processing in Taiwan ***(published)***. [1]

- **Dushan Kumarasinghe**, Nisansa de Silva "Abstract Generation with Hybrid Model Supported by Relevance Matrix" 16th International Conference on Computational Collective Intelligence in Germany ***(published)***. [2]

CHAPTER 3

LITERATURE SURVEY

Huang et al. [3] emphasized that in text summarization there are four main objectives.

1. Coverage of information
2. Information significance
3. Redundancy in information
4. Cohesion in text

As mentioned in 1.1 extractive and abstractive methods have been developed for achieving these objectives. Here we provide a background on these techniques and other relevant concepts used in the computational text summarization solutions.

3.1 Popular tools

Considering popular text summarization tools, below is a simple description of some of those tools.

We can list some of the tools such as Resoomer [4], SummarizeBot [5], SMMRY [6] TLDRThis [7] and TextCompactor [8]. Resoomer [4] is a very popular tool among college students, professors, journalists and editors. SummarizeBot [5] on the other hand is free of ads, easy to use, tool that can compress texts to save time. SMMRY [6] makes summarizations by ranking important sentences. TextCompactor [8] is also a free ad-free tool that lets users decide the percentage of the summarization size.

Resoomer [4] application creates accurate text summaries, allowing users to quickly scan publications for key subjects, find essential facts and ideas, and comprehend articles. Long content can be summarized in just 500 words for users. Anyone, including students and professors, can use this text tool to swiftly read texts, and journalists can use it to simplify information that highlights key events and filter material found in press releases.

SummarizeBot [5], With the summarization of extensive texts, an AI and blockchain-powered solution allows users to learn more while reading less. This contains whole Wikipedia articles, white papers, web pages, as well as audio and video. It compresses words to assist consumers save time when researching. It also condenses news, extracts keywords from texts, and allows users to choose summary lengths in addition to summary production. It supports English, Chinese, Russian, Japanese, Arabic, German, Spanish, French, Portuguese, etc. This uses artificial intelligence, machine learning, natural language processing and blockchain technologies for the summary generation

SMMRY [6] is programmed to condense a TXT or PDF text into the most significant sentences in just 7 seconds, making it a quick and efficient approach to grasp information. It eliminates transition words, extraneous sentences, and excessive examples using its core algorithm to provide readers with a topic-focused summary. SMMRY [6] is also used by Reddit's AutoTLDR bot to generate short summaries of long Reddit submissions by following the above-mentioned list of prescribed rules.

TLDRThis [7] is an intuitive online tool and browser extension designed to generate concise summaries of articles and lengthy texts. It simplifies the process of extracting key points by providing users with the most critical information in a short format, making it easier to quickly comprehend the main ideas without reading the entire text. This tool is especially useful for those who need to rapidly scan through online content, such as students, researchers, and professionals, by reducing long articles into digestible summaries. Its versatility and ease of use make it a go-to resource for anyone looking to save time while staying informed.

TextCompactor [8] also has a language translation function. Furthermore, it enables the option of several versions of these summarizations with the 'summarization ratio' in addition to generating text summaries. Users can vary the density of the paraphrase by changing the percentage from 5% to 80%.

3.2 Validation Mechanisms

Validating summarized text is one of the critical steps in the text summarization process, as it plays a pivotal role in analyzing the results and comparing the accuracy and performance of the model. The quality of the output is typically assessed using specific metrics that are designed to evaluate machine-generated text, often adapted from those used in machine translation tasks. However, the challenge lies in the fact that the quality of a translation or in this case, a summary is inherently subjective. There is no absolute or universally accepted measure of what constitutes a "perfect" or "excellent" summary, as interpretations of quality can vary based on individual perspectives and the intended purpose of the summary.

Due to this subjectivity, any validation metric must aim to provide scores that align closely with human quality judgments. These metrics should reflect how well the generated summary captures the essential information from the source text while maintaining coherence, fluency, and readability. Numerous mechanisms have been developed to address this, each with its own strengths and weaknesses. Some metrics focus on surface-level matching of words and phrases between the generated summary and a reference summary, while others aim to assess deeper semantic similarities.

In the following section, we will discuss some of the most popular mechanisms used for summary validation. These mechanisms are widely adopted in the field of natural language processing and provide a framework for systematically evaluating

the performance of text summarization models. By applying these metrics, we can compare model outputs against human-generated summaries and gain insights into the effectiveness of the summarization process.

3.2.1 BLEU

The Bilingual Evaluation Understudy (BLEU) metric, introduced by Papineni et al. [9], is based on the core principle that "the closer a machine translation is to a professional human translation, the better it is." BLEU is widely used for evaluating the quality of machine-generated translations and has been extended to other tasks such as text summarization. The main objective of BLEU is to provide an automated method of assessing the quality of a translation by comparing it to one or more high-quality human reference translations.

There are two fundamental components required for BLEU evaluation:

1. **Numerical translation closeness metric:** This refers to a mathematical measurement of how closely the machine-generated output matches the human reference translation.

2. **A corpus of good quality human reference translations:** These reference translations serve as a gold standard against which the machine output is compared. The more similar the machine-generated text is to these human translations, the higher the BLEU score.

The BLEU score is calculated by first computing the geometric mean of the test corpus's modified precision scores, which are based on n-grams. An n-gram represents a sequence of 'n' consecutive words, and the precision score reflects how many n-grams from the machine-generated text match the reference translations. However, BLEU adjusts the precision by introducing a "modified precision" approach, which accounts for situations where a machine translation might over-generate frequent words, thereby preventing artificially inflated precision scores.

After calculating the modified precision, BLEU applies a brevity penalty to account for translations that are too short. This penalty discourages overly short translations that could artificially match the reference by excluding important details. The brevity penalty is an exponential factor applied to the final score, ensuring that translations that are both accurate and of appropriate length receive a higher score.

The final BLEU score is obtained by taking the geometric average of the modified n-gram precisions (p_n), where p_n represents the precision for each n-gram level (e.g., unigrams, bigrams, trigrams), and then multiplying the result by the brevity penalty factor. This ensures that the score reflects not only the closeness of the translation to the reference but also its adequacy in terms of length. In this way, BLEU provides

an objective, reproducible metric for evaluating the quality of machine-generated text relative to human-generated translations.

$$\textbf{BP=}\begin{cases} 1 & if c > r \\ e^{1-r/c} & if c \leq r \end{cases} \tag{3.1}$$

$$\textbf{BLEU=BP} \cdot \textbf{exp}\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3.2}$$

3.2.2 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [10], is a widely used set of metrics designed to automatically assess the quality of summaries by comparing them to reference summaries created by humans. ROUGE is recall-oriented, meaning it focuses on capturing as much of the important content from the reference summary as possible in the generated summary. It offers various sub-metrics that measure different aspects of overlap between machine-generated and human-generated summaries, making it highly versatile for summarization tasks.

Several ROUGE metrics are commonly used in evaluation, each targeting different types of text relationships:

- ROUGE-N
  This measures the overlap of n-grams (continuous sequences of 'n' words) between the system-generated and reference summaries. The most commonly used variants are:
  - ROUGE-1: Measures the overlap of unigrams (single words) between the system and reference summaries. This is useful for evaluating basic word-level recall.
  - ROUGE-2: Measures the overlap of bigrams (pairs of consecutive words) between the system and reference summaries. ROUGE-2 provides insights into how well the system preserves word sequences and context from the reference.
- ROUGE-L
  This evaluates summaries based on the Longest Common Subsequence (LCS) . LCS measures the longest sequence of words that appear in both the system and reference summaries in the same order. ROUGE-L naturally captures sentence-level structural similarity without requiring strict n-gram matches, and it reflects how well the system maintains the flow and structure of the reference summary.

- ROUGE-W
  This is a weighted version of ROUGE-L, placing more emphasis on consecutive occurrences of the LCS. This metric rewards summaries that not only include matching sequences but also maintain them in long, uninterrupted stretches, thereby enhancing the coherence of the generated summary.

- ROUGE-S
  It evaluates co-occurrence statistics based on skip-bigrams. A skip-bigram is any pair of words that appear in the same order in both the system and reference summaries, but not necessarily adjacent to each other. This metric provides flexibility in capturing related word pairs that are important to the meaning, even if they are not consecutive.

- ROUGE-SU
  This is an extension of ROUGE-S that incorporates both skip-bigrams and unigrams. By combining the flexibility of skip-bigrams with the precision of unigrams, ROUGE-SU offers a more comprehensive measure of summary quality, balancing both content and structural relationships between the system and reference summaries.

Each of these ROUGE metrics provides different insights into how well a generated summary matches a reference summary, whether by preserving individual words, word sequences, or broader structural relationships. By utilizing a combination of these metrics, ROUGE offers a detailed and multi-faceted evaluation of summary quality, making it an indispensable tool for summarization tasks across various domains.

3.2.3 METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering)[11] is a validation metric originally developed for machine translation but is also widely applied in the evaluation of automatic text summarization. It was designed to address some of the limitations of earlier metrics like BLEU by providing a more nuanced assessment of text quality.

METEOR improves upon traditional precision-based metrics by considering factors such as synonym matching, stemming, and word order. This means that METEOR recognizes different forms of the same word and accounts for synonyms, thereby offering a more human-like evaluation of generated summaries. Additionally, METEOR incorporates a penalty for word order, assessing how well the generated summary preserves the structural alignment of the reference summary. By capturing these linguistic subtleties, METEOR offers a comprehensive measure of how well the generated summary aligns with human expectations, making it a valuable tool in both machine translation and text summarization tasks.

3.2.4 BERTScore

BERTScore [12] is a modern validation metric that leverages the power of contextual embeddings from BERT (Bidirectional Encoder Representations from Transformers) to evaluate the quality of text summarization. Unlike traditional metrics that rely on surface-level n-gram matching, BERTScore compares the cosine similarity between the embeddings of words in the generated summary and those in the reference summary. This approach allows BERTScore to capture semantic similarities and contextual nuances, offering a more sophisticated evaluation of the generated text's meaning and relevance.

BERTScore provides precision, recall, and F1 scores based on the similarity of word embeddings, giving a well-rounded assessment of the summary's quality. Due to its ability to understand context and meaning at a deeper level, BERTScore is particularly useful for evaluating abstractive summarization tasks, where preserving the essence of the original content is crucial. This makes BERTScore a powerful tool in the ongoing development and refinement of text summarization models.

3.3 Related review papers

Table 3.1 lists the papers we selected as related review papers. A comprehensive description of these papers is given below

TABLE 3.1: REVIEW OF IDENTIFIED SURVEY PAPERS ON AUTOMATIC TEXT SUMMARIZATION

Paper Title	Authors	Year	Citations
A survey on automatic text summarization	Nazari and Mahdavi [13]	2019	20
Automatic text summarization: A comprehensive survey	El-Kassas et al. [14]	2021	55
A survey on extractive text summarization	Moratanch and Gopalan [15]	2017	97
Text Summarization Techniques: A Brief Survey	Allahyari et al. [16]	2017	326
Recent automatic text summarization techniques: a survey	Gambhir and Gupta [17]	2017	466
A survey on abstractive text summarization	Moratanch and Chitrakala [18]	2016	69

Nazari and Mahdavi [13] also make an in-depth introduction to automatic text summarization and some evaluation techniques to evaluate the quality of automatic text summarization.

Referring Jones et al. [19] and Hovy et al. [20] authors emphasize the three steps

break down of the building blocks of the structural components in automatic text summarization.

1. Identification
2. Interpretation
3. Generation

In the identification phase main points and the central topics are identified and a summary is generated by gathering the important bits of the text.

In interpretation phase points identified in the first phase will be combined more cohesively. Some modifications may require in this phase. But the generated idea in this phase is more machine friendly.

The outcome of the second phase will be reformulate into a coherent new text in the generation phase.

Comparing several summarization techniques Nazari and Mahdavi [13] concludes that combining different methods into a hybrid method, we can eliminate the shortcommings and use their strengths to improve quality of the hybrid method.

El-Kassas et al. [14] have classified the summarization systems considering different aspects as shown in Figure 3.1.

El-Kassas et al. [14] emphasize the importance of developing abstractive automatic text summarization methods. The paper describes the different approaches, methods, building blocks, techniques, datasets, evaluation methods, and future research directions of summarization methods.

Referring Dutta et al. [21], El-Kassas et al. [14] claim that different algorithms produce different summaries from the same input texts and it is very promising to combine outputs from multiple summarization algorithms to produce better summaries. Also the recommendation of Mahajani et al. [22] to benefit from the advantages of both extractive and abstractive approaches by proposing hybrid automatic text summarization systems, has motivated the authors to create a comprehensive survey for researchers to enhance summary generation by combining different approaches and/or methods.

As previously mentioned, Abstractive, Extractive and combination of both named Hybrid Text summarization approaches are shown in Figure 3.4 as mentioned by El-Kassas et al. [14].

Moratanch and Gopalan [15] also provides a coherent survey on extractive text summarizaiton techniques, their challenges and popular datasets. They have categorized extractive summarization techniques as shown in Figure 3.5.

Allahyari et al. [16] focus on extractive summarization methods and provide an overview of some of the most dominant approaches in this category. They describe 3 main tasks in extractive text summarization as below.

Fig. 3.1: Text summarization classification

1. Create an intermediate representation of the input text that captures its key aspects.

2. Evaluate the sentences according to the representation.

3. Choose a summary consisting of a selected number of sentences.

When constructing intermediate representation they assign important score to each sentence. A score of a sentence represents how well the sentence explains the most important topics of the text. This score is computed by aggregating the evidence from different indicators and machine learning techniques are often used to find these indicator values.

Extractive text summarization methods have been developed more often since they

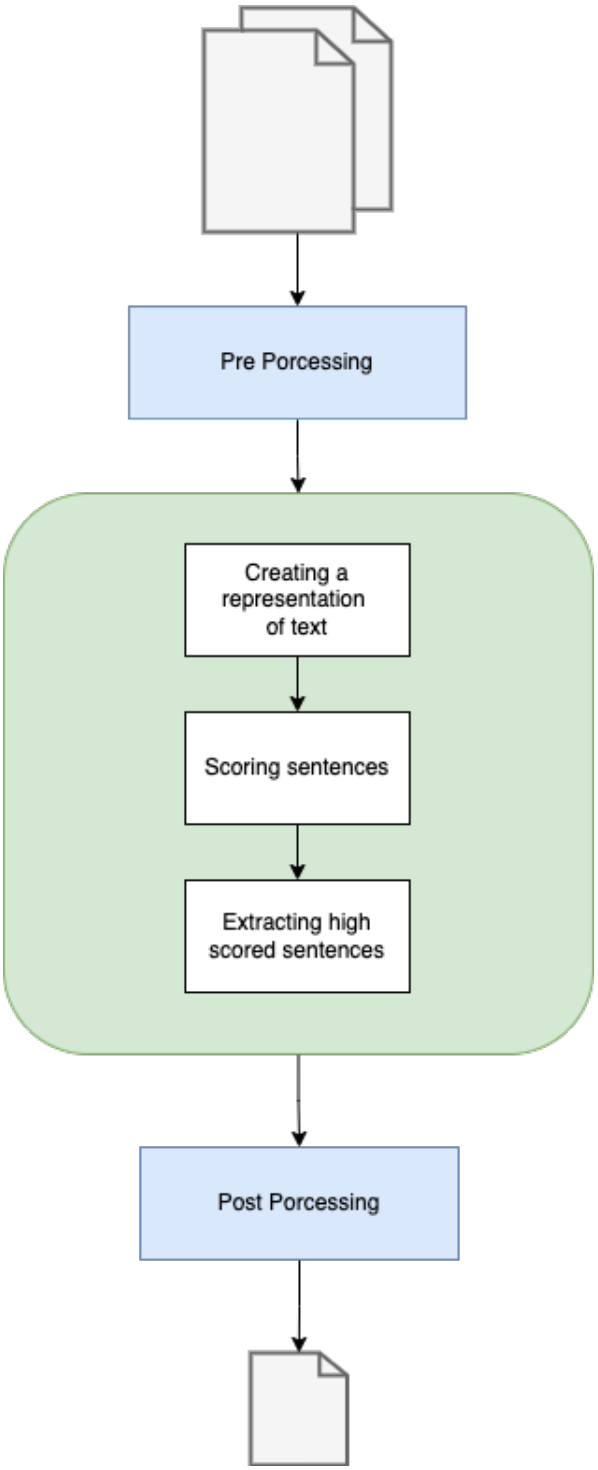

Fig. 3.2: Extractive high level architecture

are less complex than abstractive methods. Gambhir and Gupta [17] presents a comprehensive survey of recent text summarization extractive approaches developed in the last decade. A few number of abstractive and multilingual text summarization approaches also have been discussed in the paper. Their needs, advantages and disadvan-

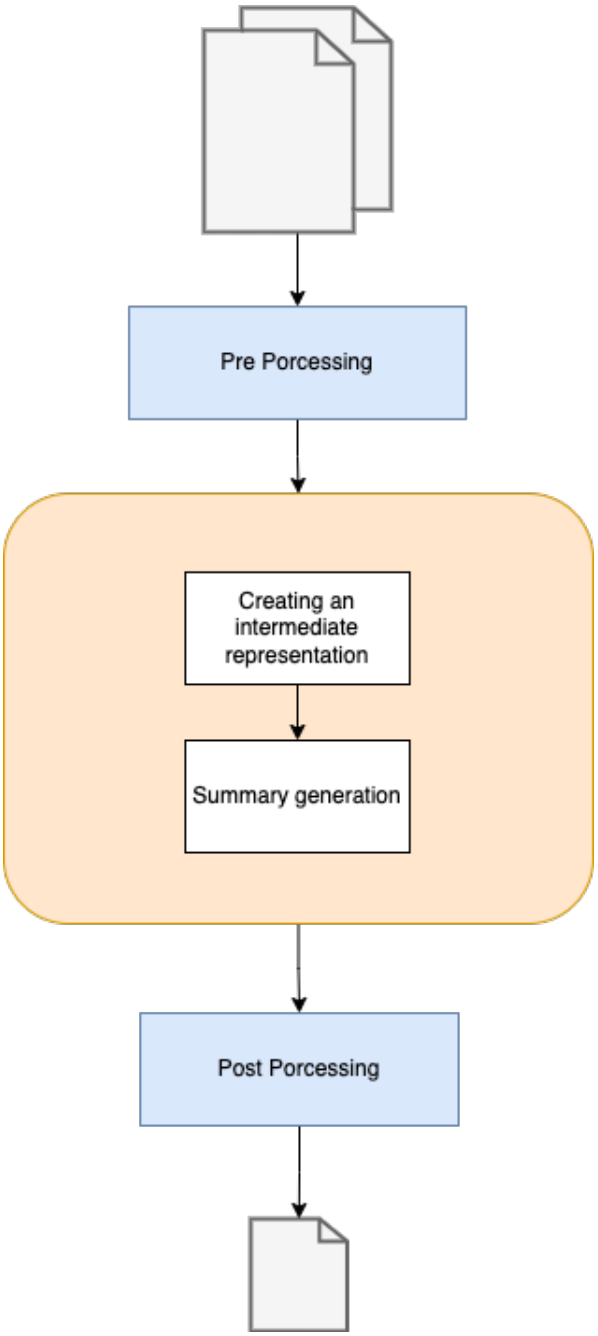

Fig. 3.3: Abstractive high level architecture

tages are identified and states the useful future directions. Moreover the authors have compared the summarization techniques against DUC 2007 [23] dataset and calculated the ROUGE [24] score as shown in Table 3.2.

Moratanch and Chitrakala [18] have done a survey on abstractive text summarization techniques, their challenges and the state of the art datasets. They claim that abstractive summarization is an efficient form if summarization compared to extractive summarization and it generates a summary that will be in more coherent form, easily

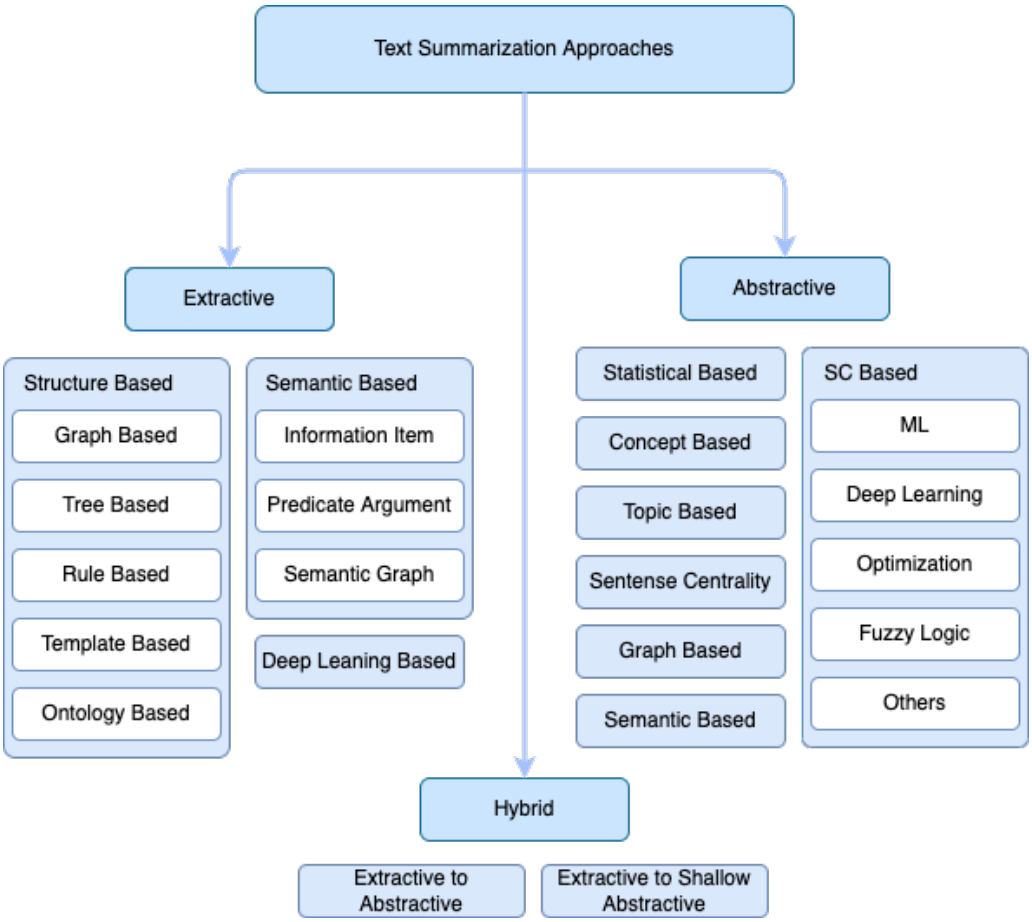

Fig. 3.4: Automatic Text Summarization Approaches with their methods

readable and grammatically correct.

Abstractive summarization can be categorized into two main types as Structure based approach and semantic based approach. Moratanch and Chitrakala [18] have depicted these two types further. categorization of the methods has been shown in Figure 3.6.

Moratanch and Chitrakala [18] note that major issue of abstractive summarization is there is no generalized framework, parsing and alignment of parse trees is difficult. Extracting important sentences, sentence ordering as in original source and information diffusion are open issues according to Moratanch and Chitrakala [18].

Jin et al. [35] present a comprehensive survey on Automatic Text Summarization (ATS), introducing a Process-Oriented Schema that aligns with real-world ATS implementations. Unlike previous surveys that categorize ATS techniques from a purely theoretical standpoint, this study structures ATS around four key components: Data Acquisition, Preprocessing, Summarization Modeling, and Evaluation Metrics. The authors highlight how the advent of Large Language Models (LLMs) has revolutionized summarization by shifting from traditional pre-training and fine-tuning ap-

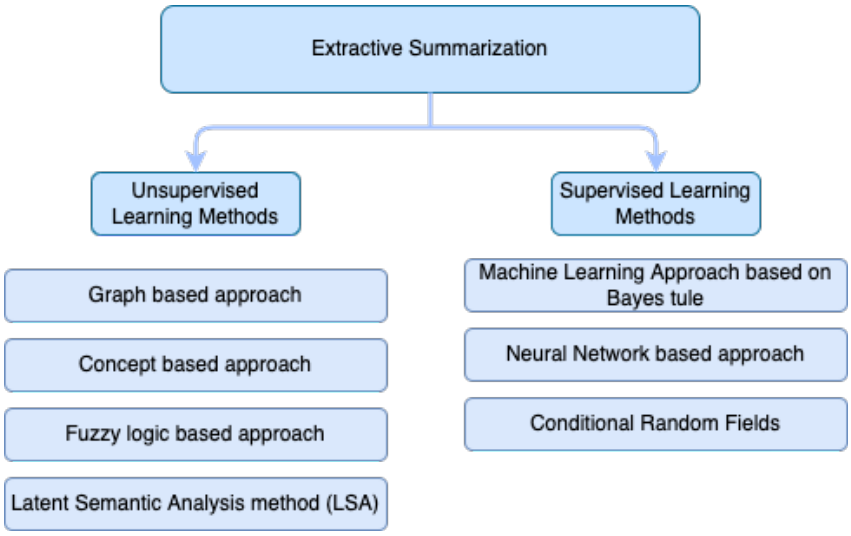

Fig. 3.5: Extractive summarization techniques

TABLE 3.2: ROUGE SCORE OF THE TEXT SUMMARIZATION METHODS ON DUC 2007 DATASET

Technique	**ROUGE-2**	**ROUGE-SU4**
Ranking-based MMR [25]	0.1262 (1)	0.1780 (1)
MCMR (B&B) [26]	0.1221 (3)	0.1753 (2)
SpOpt-comp [27] [28]	0.1245 (2)	0.1743 (3)
MCMR (PSO) [26]	0.1165 (5)	0.1697 (4)
AdaSum [29]	0.1172 (4)	0.1692 (5)
Uni + Max [30]	0.1133 (6)	0.1652 (6)
Sum_Sparse [31] [32]	0.0920 (7)	0.1460 (7)
PNR^2 [33]	0.0895 (8)	0.1291 (8)
MDS-Sparse-div [34]	0.0645 (9)	0.1167 (9)

proaches to prompt-based learning. They provide an in-depth analysis of LLM-based summarization, detailing knowledge distillation, fine-tuning, and prompt engineering techniques. The survey also reviews major open-source ATS datasets, including CNN/Daily Mail [36], XSum [37], Multi-News [38], arXiv/PubMed [39], and WikiHow [40], evaluating their suitability for different summarization tasks. Additionally, the study discusses limitations of current ATS models, such as context utilization challenges in LLMs, factual inconsistencies in generated summaries, and the computational costs associated with fine-tuning large models. By providing a structured process-oriented approach to ATS, this paper serves as a practical guide for researchers and developers working on real-world summarization applications.

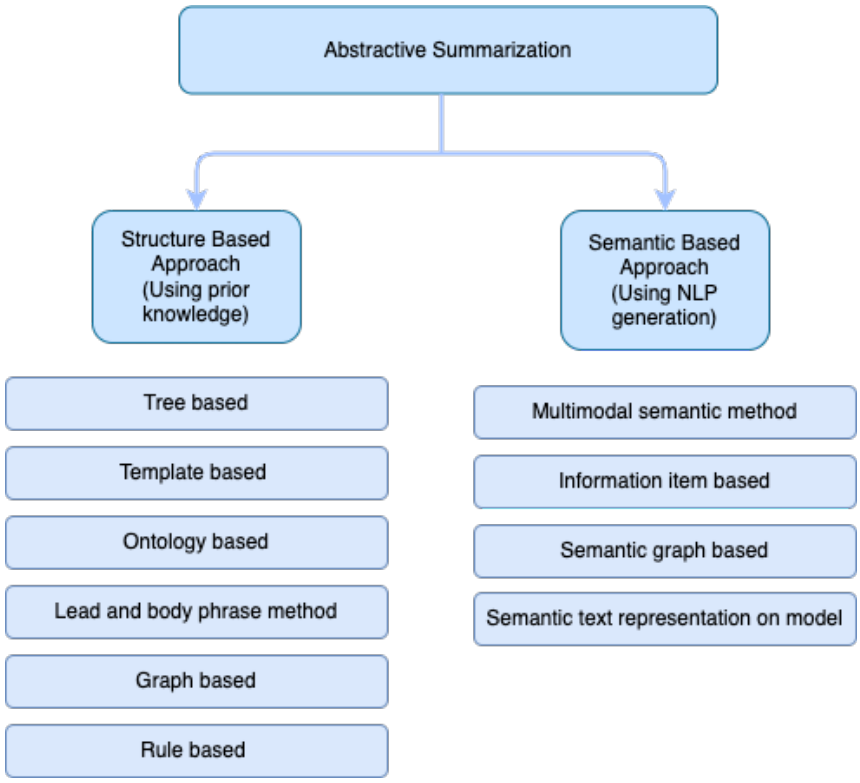

Fig. 3.6: Abstractive summarization techniques

3.4 Paper review

Text summarization has seen significant advancements with the adoption of deep learning models, particularly transformer-based architectures. Liu and Lapata [41] demonstrate how Bidirectional Encoder Representations from Transformers (BERT) [42] can be effectively applied to both extractive and abstractive summarization. Their approach introduces a document-level encoder built on BERT, which encodes sentences and stacks intersentence transformer layers to capture document-level features. For abstractive summarization, they combine the BERT encoder with a transformer-based decoder, achieving state-of-the-art performance on benchmark datasets such as CNN/Daily Mail [36], NYT [43], and XSum [44].

While Liu and Lapata [41] focus on leveraging pretrained transformers, Nallapati et al. [45] frame summarization as a sequence-to-sequence learning problem, similar to machine translation [46], speech recognition [47], and video captioning [48]. They use an attentional encoder-decoder RNN model to generate abstractive summaries, highlighting the challenge of summarization compared to translation—where summarization requires lossy compression while preserving key concepts, whereas translation seeks lossless, one-to-one alignment. Their work laid the foundation for future neural summarization approaches, including the widely used CNN/Daily Mail corpus [36], which was derived from Hermann et al. [49].

Building upon these approaches, See et al. [50] introduce a pointer-generator network to address the limitations of purely extractive or abstractive models. Their method allows the model to copy key phrases from the source document while still generating new words when necessary. This hybrid approach, aided by an attention mechanism, ensures that summaries remain both concise and coherent.

Extending these ideas, Cohan et al. [39] emphasize the importance of discourse structure in abstractive summarization, particularly for longer-form documents such as research papers. Their model processes each discourse section individually before encoding the entire document, leveraging a discourse-aware attention mechanism to prioritize critical information. Unlike traditional news-based datasets such as CNN/Daily Mail and New York Times, which contain general news summaries, scientific papers have structured discourse sections that offer better-defined summarization challenges. To support their research, Cohan et al. [39] introduced two large-scale datasets—arXiv (215,000 papers) and PubMed (133,000 papers)—which have become benchmarks for scientific document summarization.

Parallel to these developments, Shi et al. [51] conduct a comprehensive review of sequence-to-sequence architectures in abstractive summarization, evaluating various training techniques and summarization strategies. They highlight key datasets, including CNN/Daily Mail [36], Newsroom [52], and Bytecup [44], that have been pivotal in advancing research. Additionally, they present Neural Abstractive Text Summarizer (NATS), an open-source library incorporating features such as pointer-generator networks, intra-temporal attention, coverage mechanisms to mitigate repetition, and beam-search for improved generation [53].

Beyond conventional transformers, models designed for long-document summarization have gained prominence. Beltagy et al. [54] introduce Longformer, which addresses the limitations of self-attention in standard transformers. Traditional transformers struggle with processing long texts due to quadratic time and space complexity. The Longformer mitigates this issue with a sliding window attention mechanism that allows efficient processing of extended contexts while maintaining computational feasibility. Additionally, a global attention mechanism enhances its ability to capture long-range dependencies. This approach has been widely adopted for summarization tasks requiring an understanding of extended textual structures.

Recent advancements have also explored task-specific summarization, particularly in legal and domain-specific texts. Moro et al. [55] propose a transfer learning approach for summarizing legal case reports, addressing challenges in low-resource settings. Their model integrates extractive and abstractive techniques, using CNN and GRU layers for sentence extraction before generating summaries with GPT-2 [56]. Evaluations on the Australian Legal Case Reports dataset show state-of-the-art extractive summarization performance while establishing a new benchmark for abstractive summarization. Moreover, they explore cross-lingual capabilities by training models

on translations of legal cases in multiple languages, demonstrating the scalability of transfer learning in legal NLP.

Another crucial area of research involves understanding how large language models (LLMs) utilize contextual information in summarization. Ravaut et al. [57] examine the positional bias in LLMs, identifying a U-shaped performance pattern where models prioritize information at the beginning and end of a document while underutilizing content in the middle—a phenomenon termed the "middle curse." Their large-scale analysis, spanning six LLMs, ten datasets, and five evaluation metrics, reveals that simply increasing context window sizes beyond 4k tokens does not necessarily improve summarization. To address this, they propose hierarchical summarization (summarizing sections before combining them) and incremental summarization (progressively updating a summary), which show promising results for scientific papers but yield mixed outcomes in other domains.

Collectively, these studies highlight the evolution of text summarization from early RNN-based approaches to sophisticated transformer models, emphasizing key challenges such as discourse awareness, long-document summarization, and bias in LLM-generated summaries. The continued development of large-scale datasets and specialized architectures plays a crucial role in enhancing summarization quality across different domains.

3.5 Datasets

The following table provides a comprehensive overview of the state-of-the-art datasets commonly utilized in recent research, particularly in the domain of text summarization. These datasets vary significantly in terms of the number of articles they contain and the specific details they offer, such as average document length, summary length, and the context in which they were generated. The table includes datasets from widely recognized sources such as CNN/Daily Mail, New York Times, and BBC, alongside specialized collections like the arXiv and PubMed datasets. Each of these datasets plays a crucial role in advancing the development and evaluation of summarization models, offering diverse content for training and testing purposes.

TABLE 3.3: POPULAR DATASETS

Dataset	**No of articles**	**Details**
CNN/Daily Mail dataset [36]	More than 300k	On average, the source documents in the training set contain 766 words across 29.74 sentences, while the summaries comprise 53 words and 3.72 sentences
NYT [43]	More than 1.8 million	Articles published by the New York Times from January 1, 1987, to June 19, 2007
XSum [37]	More than 220k	consists of BBC articles and accompanying single sentence summaries
Newsroom dataset [52]	1.3 million (1.2 million publicly available)	This contains 1.3 million articles and summaries written by authors and editors in the newsrooms of 38 major publications
Bytecup dataset [44]	1.3 million (1.1 million released for training)	The Byte Cup 2018 International Machine Learning Contest released a new dataset, commonly known as the Bytecup dataset
arXiv Dataset [39]	215K	Avg. doc length=4938 words; Avg. summary length=220 words
PubMed Dataset [39]	133K	Avg. doc length=3016 words; Avg. summary length=203 words
Multi-News [38]	56K	Avg. doc length 2,103 words; Written by professional journalists
WikiHow [40]	230K	Avg. doc length 579 words; Avg. summary length 54 words; Human generated summaries of WikiHow.com articles

CHAPTER 4

METHODOLOGY

In this section, we provide a comprehensive overview of the research process that underpins our work. We begin by detailing the generation of our dataset, a critical step that involved collecting and curating large-scale, high-quality data to ensure the robustness and applicability of our research findings. The selection and preparation of this dataset were guided by specific criteria aimed at capturing a diverse and representative sample of the documents relevant to our study.

Following dataset generation, we move on to the evaluation of pre-summarization methods. This step is crucial as it involves assessing various existing summarization techniques to determine their effectiveness and suitability for our specific research goals. By conducting a thorough evaluation, we identify the strengths and limitations of these methods, which inform the subsequent development of our own approach.

Next, we delve into the relevance evaluation, which examines how well different sections of a document contribute to the final summary. This involves analyzing the importance and influence of each section, such as the introduction, related work, methodology, results, and conclusion on the overall summary. Understanding the contribution of each section helps us refine our model to produce more accurate and contextually appropriate summaries.

We then describe the structure of our model, which has been carefully designed to handle the complexities of summarizing long-form documents. The architecture of the model is discussed in detail, highlighting the innovative features and components that differentiate it from existing models. This includes the specific layers, attention mechanisms, and other architectural choices that contribute to its performance.

Model tuning is another critical aspect covered in this section. We explain the process of fine-tuning our model to optimize its performance. This involves adjusting various hyperparameters, such as learning rates and batch sizes. The goal of this tuning process is to ensure that the model generalizes well across different datasets and produces high-quality summaries consistently.

Finally, we outline the prediction flow of our model, describing the sequence of operations that occur when the model is used to generate summaries. This includes the input processing and the generation of the final summary. By providing a clear explanation of this flow, we aim to offer insights into how the model operates in practice and how each component contributes to the overall performance.

Overall, this section serves as a detailed roadmap of our research process, guiding the reader through each critical step from dataset generation to the final prediction, and highlighting the key decisions and methodologies that have shaped our work.

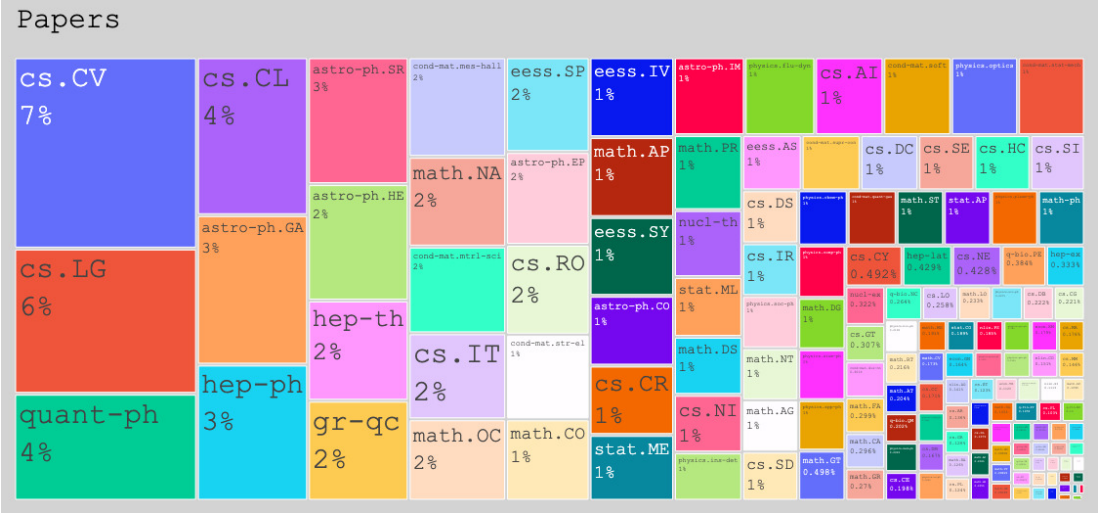

Fig. 4.1: Research paper primary tag (**arXiv** tag) portions in the dataset

4.1 Dataset

As Kreutzer et al. [58] observes, the quality of the datasets used plays a vital role in the ultimate outcome of any research.

In this initiative, we are unveiling an innovative and meticulously crafted dataset, organized section-wise, sourced from the renowned repository of scholarly works, **arXiv**[1]. This dataset encompasses a wealth of contemporary research papers, amassed from **arXiv**, totaling more than 350k documents. What sets this dataset apart is the thorough separation of each research paper into its constituent sections, enhancing the versatility and depth of analysis.

For the convenience and flexibility of researchers and practitioners, we present both cleaned (301k) and raw (350k) versions of these research papers within the dataset. The cleaned versions have undergone careful editing and formatting to ensure clarity and consistency, facilitating seamless utilization for diverse research purposes. In contrast, the raw versions maintain the authentic original content, providing a glimpse into the unaltered nature of the scholarly documents.

To aid in the navigation and categorization of this extensive repository of knowledge, each research paper within our dataset is enriched with **arXiv** tags, systematically assigning papers to specific domains or subject areas. This categorization offers a structured approach, allowing for refined searches and analyses based on the field of interest or specialization.

In Figure 4.1, we present an illustrative visualization showcasing the distribution of primary tags across the research papers, providing a snapshot of the dataset's domain landscape. This visual representation offers valuable insight into the diversity

[1] https://arxiv.org/

and breadth of topics covered within our dataset, aiding researchers in identifying and exploring research papers relevant to their areas of focus.

4.1.1 Dataset Generation

This section outlines the steps involved in the dataset generation process, detailing each stage from the initial data collection to the final compilation of a structured and analyzable dataset. The dataset generation process was a multi-phase procedure that required careful planning and systematic execution to ensure the quality, consistency, and relevance of the data used for analysis. Starting with the extraction of raw data from academic sources, each subsequent step involved refining, organizing, and formatting the data to meet the specific requirements of our research objectives.

The dataset generation process included tasks such as identifying relevant academic papers, extracting key metadata, retrieving the full text and abstract sections, and processing raw LaTeXfiles to make them suitable for analysis. By following these steps, we were able to build a comprehensive and well-structured dataset that formed the foundation of our study. This section will provide a detailed breakdown of each of these tasks, illustrating how the various components of the dataset were collected and prepared for use in our experiments.

4.1.1.1 Paper Data Extraction from arXiv

The initial phase of our dataset generation process involved systematically extracting relevant data from academic papers hosted on arXiv.org. To facilitate this, we developed a custom web crawler specifically designed to identify and process papers authored in LaTeXformat. This choice of format was crucial, as LaTeXis widely used in academic fields such as computer science, mathematics, physics, and other technical disciplines due to its capability to handle complex formatting, equations, and references. By targeting papers formatted in LaTeX, we ensured that the data we collected would be both rich in structure and standardized for further processing.

Our web crawler was designed to systematically navigate the arXiv.org repository, starting with the identification of relevant papers. It first scanned the site to gather a list of papers, filtering for those authored in LaTeX. This was followed by the extraction of essential metadata for each paper, including the title, author names, unique arXiv identifiers, publication date, and source URLs. These metadata elements formed the backbone of our dataset, creating a linkage between each paper's identity and its corresponding location on the web. By maintaining this connection, we ensured that each entry in our dataset was traceable and could be directly accessed for future reference.

Once the metadata was collected, the next task focused on extracting the abstract sections from the identified papers. The abstract, being a concise summary of the

research, provided a valuable overview of the paper's content, objectives, and findings. Since abstracts are typically displayed on the paper's main page on arXiv.org, our crawler was programmed to extract this information and store it as part of our dataset. This extraction was done in a consistent format to facilitate uniformity across all collected abstracts, allowing us to later process and analyze them efficiently.

In addition to the abstracts, our crawler was designed to extract the source LaTeXfiles associated with each paper. These files, often accessible through the download links provided on arXiv, were critical for our subsequent analysis and processing steps. By retrieving the raw LaTeXsource files, we gained access to the full content of the papers, including figures, tables, equations, and references, all of which are often essential for comprehensive analysis. The source files were downloaded and archived for future use, ensuring that we could return to the original content as needed for any detailed study or extraction beyond the abstract level.

This initial data extraction phase was foundational to our overall research process. It allowed us to compile a comprehensive dataset composed of detailed metadata, abstracts, and raw LaTeXsource files. This dataset served as the basis for the later stages of our research, where we applied various processing techniques to analyze, summarize, and interpret the academic content. The systematic extraction of paper data from arXiv.org ensured that we had a reliable and rich corpus of academic papers to work with, providing the necessary input for our subsequent experiments and analyses.

4.1.1.2 Retrieval and Processing of LaTeX Source Files

After extracting the abstracts, the next critical step in our workflow was downloading and unzipping the LaTeXfiles to ensure that we had access to the most detailed and manipulable version of the research paper's content. These source LaTeXfiles are typically rich in information and structure, making them ideal for further analysis and manipulation. By working directly with these files, we were able to maintain the integrity of the original content while preparing it for processing.

Once the source files were securely obtained, the subsequent task involved transforming them into a format that could be efficiently analyzed and processed by our system. This transformation was necessary because raw LaTeXfiles, while highly structured, contain a significant amount of markup and formatting that is not immediately suitable for direct use in machine learning or text analysis models. To handle this conversion, we utilized the pylatexenc library[2], a robust and versatile tool specifically designed for parsing and interpreting LaTeXdocuments.

With pylatexenc, we were able to generate a node tree representation of each document. This node tree acted as a structured hierarchical representation of the LaTeXdocument, breaking down the content into manageable nodes that could be in-

[2] https://pylatexenc.readthedocs.io/

dividually parsed and analyzed. The node tree format provided a clear and organized view of the document's structure, making it easier to extract meaningful information from the various sections, such as the abstract, figures, tables, and references, while disregarding unnecessary markup and formatting tags.

By employing this systematic approach, we ensured that the raw LaTeXcontent was translated into a format that was not only usable but also retained the rich structural and semantic details of the original research paper. This structured representation laid the foundation for further processing steps, enabling us to analyze and manipulate the content efficiently as part of our overall methodology.

4.1.1.3 Creation and Utilization of Node Trees

The node tree is a hierarchical structure that represents the full composition of a LaTeX document, with each node corresponding to a distinct element of the document such as sections, subsections, paragraphs, and even smaller components like equations, figures, tables, and citations. This structured representation was essential for navigating through the document in an organized and systematic way, enabling precise identification and extraction of various sections of interest.

By leveraging the node tree, we were able to isolate specific parts of the research paper, including the introduction, methodology, results, and conclusions. Each of these sections, being key components of academic papers, held unique value for our analysis. This granular level of extraction allowed us to capture the content in its most natural form, ensuring that each part of the paper could be treated and analyzed independently. The node tree's structure also enabled us to preserve the relationships between different parts of the document, which was critical for maintaining context when analyzing interconnected sections like methodologies tied to specific results.

This detailed representation not only facilitated the segmentation of documents but also contributed to building a rich, structured dataset. Each component of the paper, whether it was a section or a smaller element like an equation or figure caption, could be extracted as a discrete unit, which we could then analyze separately or in relation to other parts of the text. This made it possible to perform targeted analysis on specific sections, such as focusing on the methodologies or comparing the results across different papers, enhancing the overall depth and quality of the dataset.

Furthermore, the node tree structure allowed us to identify patterns and relationships within the text, such as recurring terminology or structural similarities across papers, which supported our broader research objectives. By observing these patterns, we were able to refine our approach and improve the ways in which we handled, categorized, and analyzed academic documents. The ability to systematically process and interpret these patterns helped us gain insights into the structure and content of academic papers, aiding in the generation of high-quality textual outputs.

Overall, this methodical approach to dataset generation, starting from crawling arXiv.org to extracting and processing LaTeXfiles using node tree representations, allowed us to create a comprehensive and highly detailed dataset. This dataset not only preserved the integrity of the original documents but also provided the structure needed for in-depth analysis, making it a crucial component of our research methodology.

4.1.2 Cleaned Dataset

As a foundational component of this research, we created and worked with a carefully curated, section-divided cleaned dataset, referred to here as the cleaned dataset. This dataset forms the backbone of the research paper summarization process, as it provides a well-organized and structured source of data that is optimized for both training and testing purposes. By dividing the research papers into distinct sections such as the introduction, related work, methodology, results, and conclusion, the dataset allows for more precise and targeted summarization efforts. This section-wise organization is critical because it aligns with the structure of research papers, ensuring that the most important and relevant information from each section is captured and reflected in the generated summaries.

The cleaning process involved multiple steps aimed at enhancing the quality and usability of the dataset. Initially, raw research papers often contain noise, such as irrelevant metadata, figures, tables, and references that may not contribute directly to the core content of the paper. To address this, we implemented a series of data preprocessing techniques, including the removal of unnecessary elements, the normalization of text (e.g., ensuring consistent capitalization and formatting), and the correction of any errors or inconsistencies in the raw data. Additionally, the dataset was thoroughly checked for missing or incomplete sections, ensuring that each paper included in the dataset was fully intact and correctly labeled.

Once the dataset was cleaned and divided into sections, it was made available for broader use within the research community. To facilitate further research and experimentation, we are publicly releasing both the dataset and the source code for the entire data extraction and cleaning process. The dataset, along with the source code[3], is hosted on GitHub and can be accessed publically. This repository contains the tools and scripts necessary to reproduce the cleaning process, as well as to apply the same extraction techniques to other datasets. By making this resource available, we aim to support future research efforts that require clean, well-structured datasets for tasks such as summarization, natural language processing, and other machine learning applications.

The decision to release the dataset and source code serves several key purposes. First, it encourages transparency and reproducibility in research by allowing others to

[3] https://github.com/agia-research/data_collector

verify and build upon the work done in this study. Second, it provides a valuable resource for other researchers who may wish to expand the dataset or adapt the extraction process to suit different domains or tasks. The open availability of this dataset enables the creation of additional datasets that follow the same section-wise partitioning and cleaning methodology, making it easier to conduct comparative studies across different domains or use cases.

Moreover, this cleaned dataset is designed to be flexible and adaptable. Researchers can utilize the provided source code to further process the dataset as needed, tailoring it to specific needs or research objectives. For example, one might wish to apply additional filtering criteria to focus on specific types of papers or research fields, or to modify the structure of the dataset to align with the requirements of a particular machine learning model. The modularity of the extraction process, combined with the availability of the code, ensures that the dataset can be easily customized for various applications.

In summary, the cleaned dataset is a crucial asset for this research, providing a structured and reliable source of data for summarization tasks. Through the public release of both the dataset and the associated extraction process code, we aim to contribute to the broader research community, fostering collaboration, transparency, and the further development of high-quality datasets for natural language processing and machine learning. This initiative opens the door to future research and dataset creation efforts, empowering researchers to explore new avenues in summarization and beyond.

4.1.3 Raw Dataset

In addition to the cleaned and section-divided dataset, we recognize that researchers often have different requirements depending on the nature of their studies. For some tasks, it is essential to have access to unprocessed or unfiltered data, allowing researchers to apply their own cleaning procedures or extract specific sections based on their unique objectives. To address these needs, we are also providing the raw dataset, which consists of unprocessed LaTeXsource texts extracted from the **arXiv** repository.

The raw dataset contains the original LaTeXsource code files, including the full content of the research papers as they were published on arXiv. This includes all the sections, figures, tables, equations, references, and metadata that were part of the original tex files. By making this raw data available, we give researchers the flexibility to apply their own preprocessing techniques, such as removing irrelevant sections (e.g., references, appendices, or figures) or extracting specific sections like the Introduction, Methodology, or Results for their customized research tasks.

Providing the raw dataset in this form offers several key advantages. First, it enables researchers to work with the complete text of a paper, allowing them to clean and organize the data according to their specific requirements. For example, some re-

searchers might be interested only in the methodology and results sections for certain types of analysis, while others may need to focus on the introduction or related work sections for different tasks. Having access to the raw dataset ensures that users have the flexibility to tailor the dataset to their exact needs.

Second, by making the original LaTeXsource files available, we also provide a more transparent view of the structure and formatting of the research papers. LaTeXis a widely used format in academic publishing, particularly in fields such as computer science, mathematics, and physics. Many researchers working in these fields are familiar with LaTeXand can easily manipulate the raw source code to extract or clean the content as needed. This raw dataset allows for more direct control over the data cleaning and extraction process, making it particularly useful for studies where precise handling of specific sections is required.

Moreover, the raw dataset preserves all elements of the original research papers, including mathematical formulas, citations, figures, and bibliographies, which are often stripped away in the cleaned datasets. This ensures that researchers who may need access to these specific components for their analyses or experiments can retain them during the cleaning process. For example, a researcher interested in analyzing the mathematical formulations across a collection of papers would benefit from the original LaTeXfiles, which include all equations and expressions in their native format.

We are making the raw dataset openly available so that researchers can integrate it into their own data pipelines, preprocess it as necessary, and utilize it for a range of applications, including text mining, summarization, topic modeling, or even citation analysis. The raw dataset can also be a valuable resource for researchers developing new preprocessing algorithms, as it provides a rich and diverse set of research papers with varying structures, content types, and formatting styles. This makes it an ideal dataset for testing and refining data cleaning techniques.

The raw LaTeXsource texts can be accessed in their entirety, offering researchers full control over how they manipulate the data. Whether the goal is to replicate the cleaning process we performed for the cleaned dataset or to apply a completely different method, this raw dataset serves as a versatile and flexible resource for a wide variety of academic and machine learning tasks.

In conclusion, the raw dataset provides an unprocessed version of the research papers, giving researchers the flexibility to clean, structure, and analyze the data according to their specific needs. By releasing this dataset along with the cleaned dataset, we aim to offer a comprehensive resource for both general and customized research workflows, facilitating a wide range of studies that depend on clean, structured, and flexible datasets.

4.2 Approach

We have selected GPT-Neo [59] as the primary language generation model for this study due to its lightweight architecture and flexibility, making it well-suited for use in resource-constrained environments without sacrificing performance. As illustrated in Figure 4.2 and Figure 4.3, the workflow for utilizing GPT-Neo involves a series of fine-tuning steps followed by the text prediction process, which enables the model to adapt to specific tasks such as abstract generation. This adaptability was a crucial factor in choosing GPT-Neo, as it allows for efficient experimentation in environments where computational resources are limited.

To optimize the performance of GPT-Neo in generating high-quality abstracts, we conducted a structured set of experiments aimed at exploring the impact of different input strategies. These experiments were designed to test the effectiveness of various pre-summarization techniques, where the input data fed into the model was pre-processed in diverse ways to improve the clarity, coherence, and relevance of the final outputs. By experimenting with different input formulations, we aimed to systematically identify which strategies produced the most coherent and well-structured abstracts, reducing the need for post-editing and improving the overall utility of the generated content.

Our methodology included iterative testing, where each input strategy was carefully evaluated based on the quality of the abstracts generated. This approach enabled us to fine-tune not only the model but also the pre-summarization techniques, ensuring that the language model was capable of producing abstracts that closely matched the intended outputs in terms of both content accuracy and linguistic fluency. Additionally, the iterative nature of these experiments allowed for continuous refinement, where feedback from earlier stages was used to inform subsequent testing cycles, ultimately leading to an optimized approach for abstract generation using GPT-Neo.

By leveraging this systematic approach, we were able to push the boundaries of what GPT-Neo can achieve in the context of abstract generation, demonstrating that with appropriate input strategies, even a lightweight model can deliver high-quality results in a resource-efficient manner.

One of the key constraints encountered during the model training process was the limitation on the maximum chunk size, which was capped at 2048 tokens. This token limit created a significant challenge, as it restricted the amount of text that could be processed by the model at any given time. Research papers, being inherently lengthy and complex documents, often exceed this limit, making it difficult to include all relevant information within a single input. As a result, it became necessary to carefully manage and allocate the available tokens to different sections of the input data, including the abstract, main text, and associated tags.

The division of the 2048-token limit between these components is illustrated in Figure 4.4, where a portion of the token allocation was dedicated to the abstract, an-

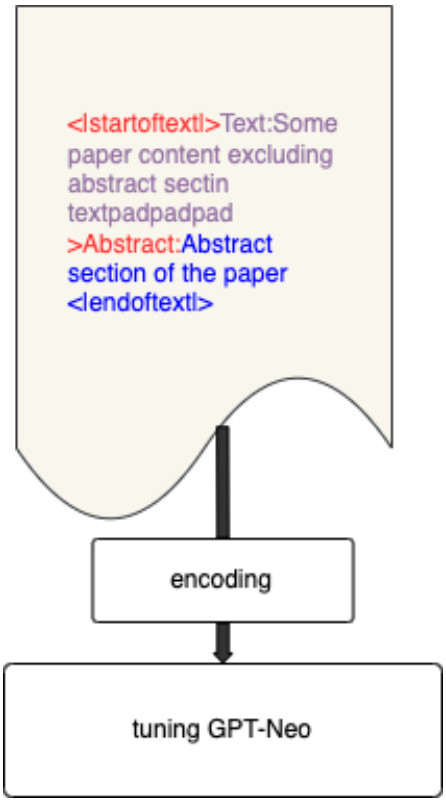

Fig. 4.2: Tuning GPT-Neo for abstract generation

other portion to the body text, and the remainder to the tags. This balancing act was crucial to ensure that the most important information from each section of the research paper was included, while still adhering to the token limit. However, this constraint also introduced an additional layer of complexity, as it required us to prioritize certain sections of the document over others, depending on their relevance to the task at hand.

To address this limitation, we had to experiment with different strategies for token allocation, ensuring that the essential information from each section was retained while minimizing the risk of truncating important content. This required careful consideration of which parts of the text would have the greatest impact on the model's ability to generate accurate and coherent outputs. Balancing the token limit across these sections proved to be a critical factor in the success of the model training process, as it directly influenced the quality and comprehensiveness of the final results.

Determining the appropriate number of tokens, denoted as N, for the abstract section was a critical step in optimizing the model's performance. This decision could not be made arbitrarily, as it directly impacted the model's ability to effectively process and generate coherent abstracts. To make this determination in a data-driven manner, we analyzed the token size distribution of the abstract sections within our dataset. This analysis, illustrated in Figure 4.5, provided insights into the natural distribution of token lengths across the abstracts, allowing us to make an informed decision.

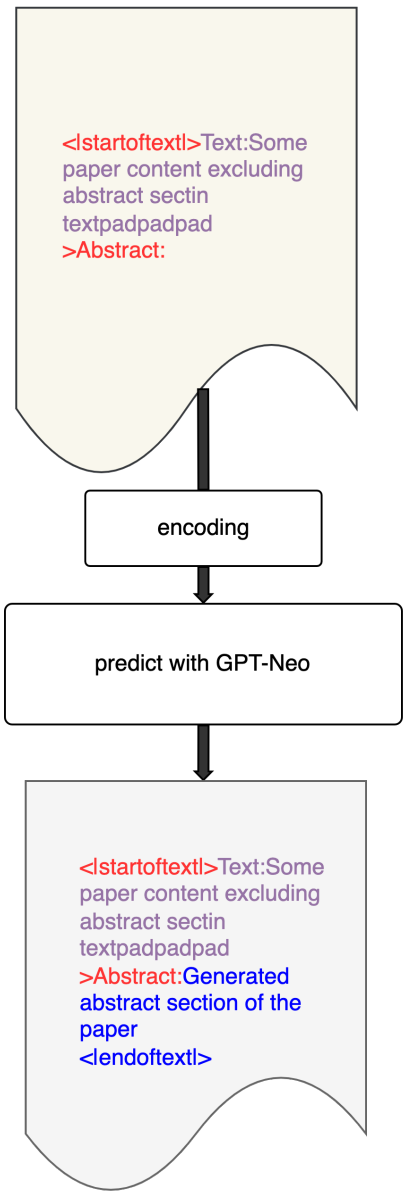

Fig. 4.3: Predicting abstracts with GPT-Neo

By closely examining this distribution, we identified key statistical markers, specifically focusing on the third quartile boundary. The third quartile, which represents the upper 3rd quartile of the token distribution, was particularly important because it gave us a reasonable estimate of the maximum number of tokens that could encapsulate most abstracts without exceeding the token limit or truncating vital information. Based on this examination, we selected 185 tokens as the desired value for $N$the number of tokens allocated for the abstract section during both the training and prediction phases.

This data-driven approach ensured that our selection of 185 tokens was grounded in the actual characteristics of the dataset, providing a balance between including sufficient information in the abstract while staying within the constraints of the token limit.

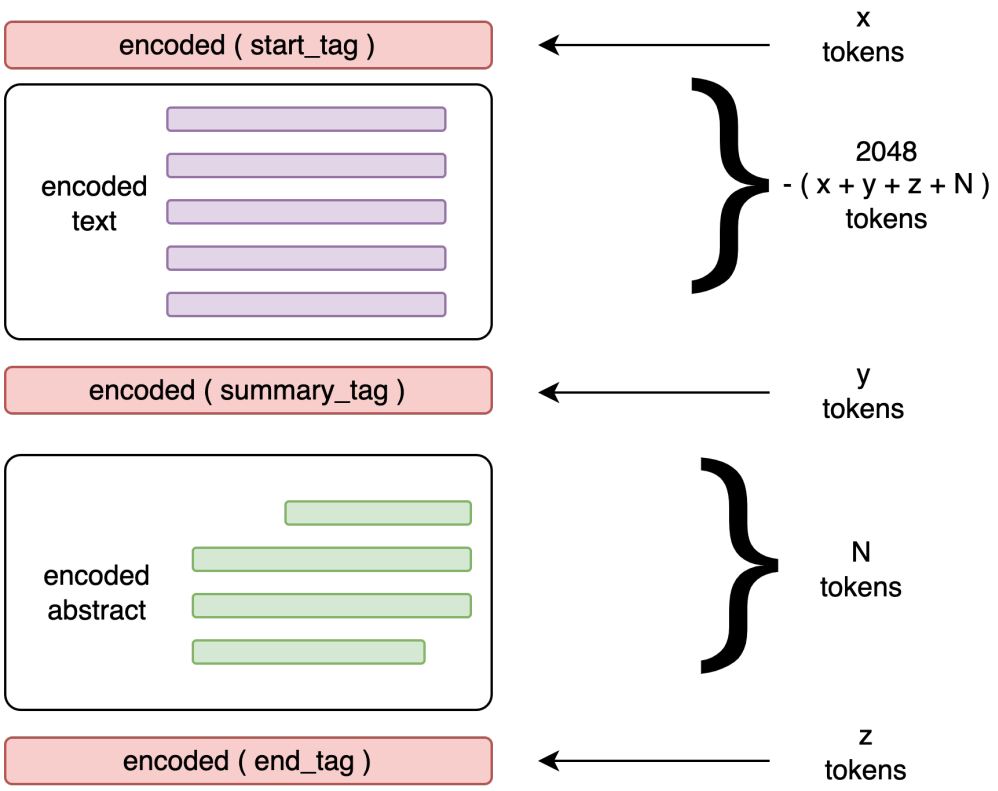

Fig. 4.4: Token size portions for GPT-Neo model feeding

By aligning N with the token distribution observed in the dataset, we aimed to maximize the quality and representativeness of the abstracts used in the formatted text for training and prediction, ultimately contributing to more accurate and coherent model outputs.

After determining the value of N for the abstract section, the next step was to calculate the size of the text body that could be included within the 2048-token constraint. This process involved reserving the first N tokens specifically for the abstract, ensuring that the abstract was always accounted for within the token limit. Additionally, tokens had to be allocated for other essential components, such as the *start*, *summary*, and *end* tags, which play a crucial role in structuring the input data for the model. These tags were assigned token lengths denoted by x, y, and z respectively.

To calculate the number of tokens available for the body text, we used the equation $2048 - (x + y + z + N)$, where x, y, and z represent the token lengths of the tags and N is the number of tokens reserved for the abstract. This formula provided us with the remaining number of tokens that could be allocated to the main body text. However, given the typical length and complexity of research papers, this full allocation of 2048

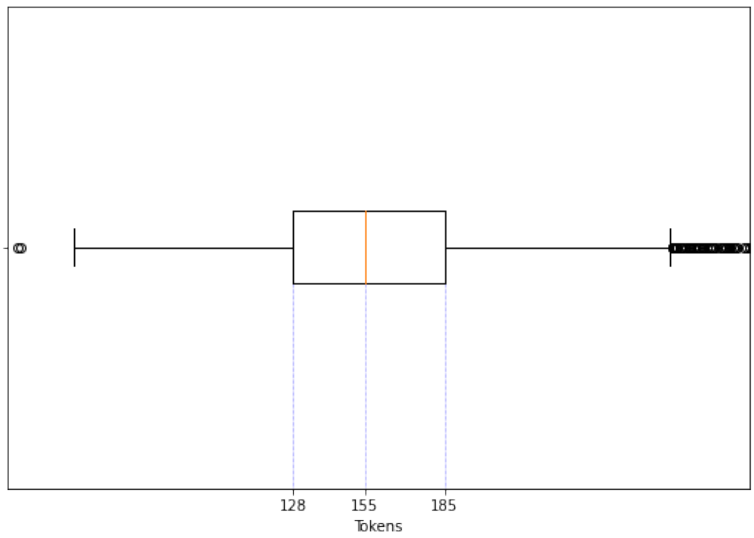

Fig. 4.5: Token size distribution of abstract sections

tokens often proved insufficient to cover the entire content of a paper.

To address this limitation, we implemented a pre-summarization strategy. This technique involved condensing the body text of the research papers in a way that retained the most relevant and critical information, ensuring that the text could fit within the available token window, referred to as the **Body Size**. By applying pre-summarization, we were able to limit the body text to the calculated token size, ensuring that the model received a manageable input length without sacrificing key content necessary for generating accurate and coherent outputs.

4.3 Pre-Summarization

To effectively utilize the available space and ensure that the text fits seamlessly within it, we needed to carefully select which portions of the text would be used as inputs for the GPT-Neo model. This selection process was crucial because the quality of the generated output heavily depends on the relevance and conciseness of the input provided to the model.

To achieve this, we employed a pre-summarization step, where we meticulously reviewed and condensed the original text. This step involved identifying the most critical information and distilling it into a more compact form. By summarizing the content beforehand, we were able to ensure that only the most pertinent and essential details were included as inputs for GPT-Neo. This approach not only maximized the efficiency of the text generation process but also helped in producing high quality abstracts that were both informative and well aligned with the original content.

Two main mechanisms were tested for pre-summarization:

1. Vector average method
2. Extractive method

These approaches for converting long text into a trainable or predictable vector are shown in Figure 4.6. After the text is reduced, it is encoded and formatted with predefined tags.

In the **Vector average method**, the research paper text, excluding the abstract, is divided into chunks of **Body Size** and converted using the GPT-2 Tokenizer [56], followed by average pooling. This results in a vector of 2048 tokens, with the first N tokens representing the abstract, followed by the flag tokens, and the average pooled context of the rest of the paper.

The **Extractive method** selects the maximum number of sentences that fit within the token limit, preserving the original meaning. This method employs following algorithms.

1. Text Rank [60]
2. Lex Rank [61]
3. Latent Semantic Analysis (LSA) [62]

Once the text is limited to the **Body Size**, it is converted to *tfrecords*, supporting distributed datasets and leveraging parallel Input/Output (I/O) . Each paper's LaTeXsource is encoded with predefined *start*, *summary*, and *end* tags to guide the model in text prediction.

The evaluated results of the pre-summarization methods are thoroughly detailed in Section 5.0.1, where we provide an in-depth discussion of the strengths and weaknesses of each approach. During the evaluation process, we focused on the performance of various pre-summarization algorithms in combination with the GPT-Neo transformer model. The goal of this evaluation was to determine which method most effectively enhanced the quality, coherence, and overall relevance of the abstracts generated by GPT-Neo. This required careful consideration of several factors, including the ability of each method to accurately condense key information while maintaining the contextual integrity of the original research paper.

Our evaluation involved a comparative performance analysis, in which we scored each pre-summarization method based on its ability to provide a structured and meaningful input for GPT-Neo. Among the methods we tested, Latent Semantic Analysis (LSA) [62] emerged as the most effective. LSA, a well-known technique in natural language processing, uses statistical methods to identify relationships between words and concepts within large bodies of text. Its ability to distill the most semantically

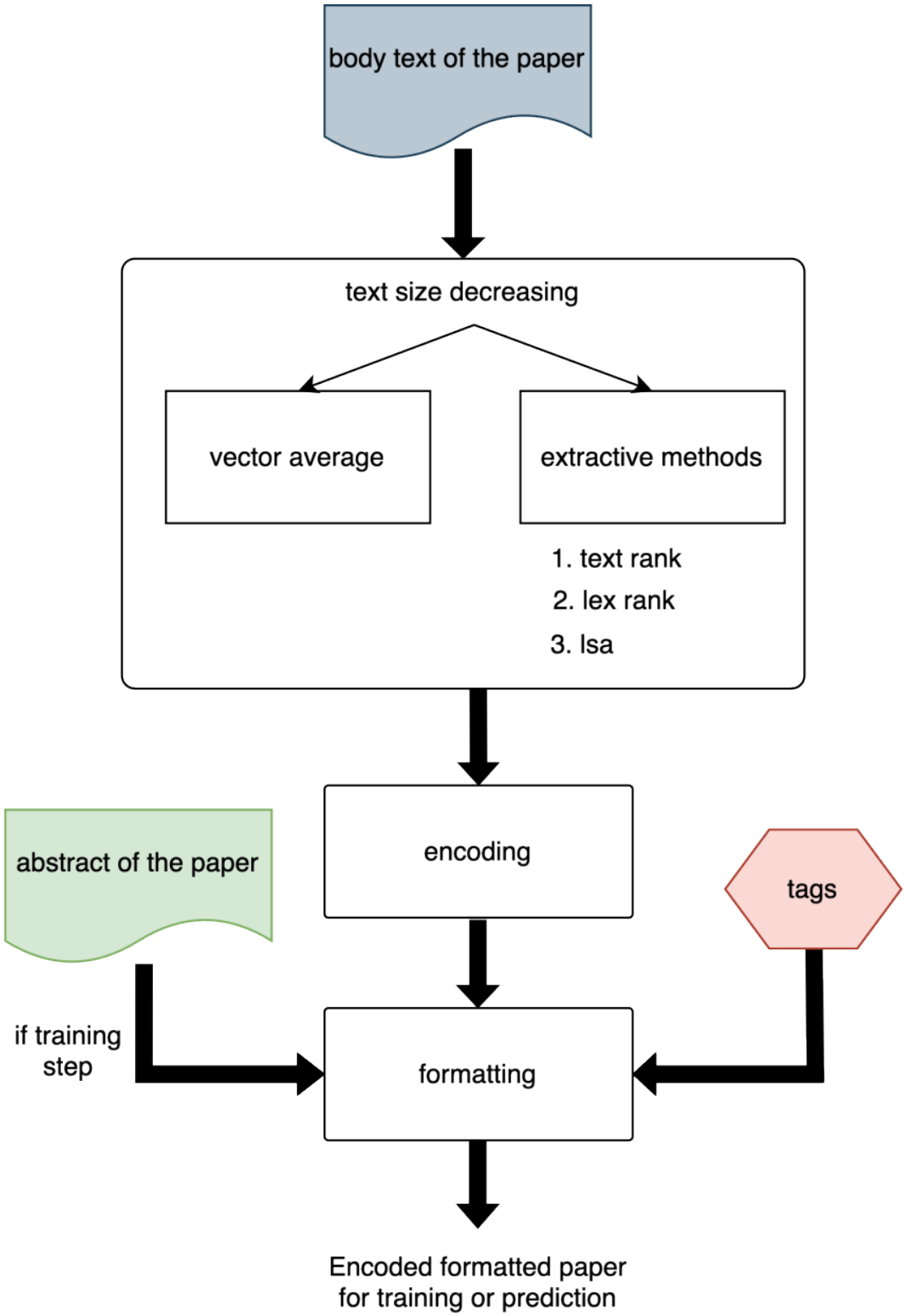

Fig. 4.6: Data preparation overview

relevant information from the various sections of a research paper made it an ideal candidate for generating coherent and contextually accurate abstracts.

LSA's effectiveness in this context can be attributed to its core capability of identifying latent patterns in text, which allows it to capture the underlying themes and key concepts of a paper with minimal loss of important information. When paired with GPT-Neo, LSA provided a pre-summarized input that was not only concise but also structurally sound, which in turn improved the quality of the output generated by the model. The abstracts produced through this approach were noticeably more coherent and contextually aligned with the original content of the research papers compared to other pre-summarization methods.

One of the key reasons we selected LSA as our preferred pre-summarization approach for this study is its ability to strike a balance between simplicity and depth. Unlike more complex algorithms that may overcomplicate the summarization process, LSA's straightforward statistical foundation allowed it to efficiently distill important information without introducing unnecessary noise or errors into the input. This simplicity, combined with its powerful semantic analysis capabilities, made it the optimal choice for our study, especially when dealing with a wide variety of research papers spanning multiple academic disciplines.

Furthermore, the results demonstrated that LSA consistently outperformed other pre-summarization algorithms in terms of the accuracy of the extracted information and the logical flow of the abstracts generated by GPT-Neo. While other methods may have captured key points, they often failed to maintain the same level of clarity and contextual relevance. In contrast, LSA's ability to identify key semantic relationships between different parts of the text allowed it to generate inputs that contributed to more meaningful and coherent outputs.

In conclusion, based on the thorough evaluation and comparison of various pre-summarization methods, Latent Semantic Analysis was selected as the most effective and reliable approach for this study. Its ability to enhance the performance of GPT-Neo by providing a well-structured and semantically rich input made it an invaluable tool in our research, leading to more accurate, coherent, and contextually appropriate abstract generation. As a result, LSA was chosen as the preferred pre-summarization method moving forward in this study.

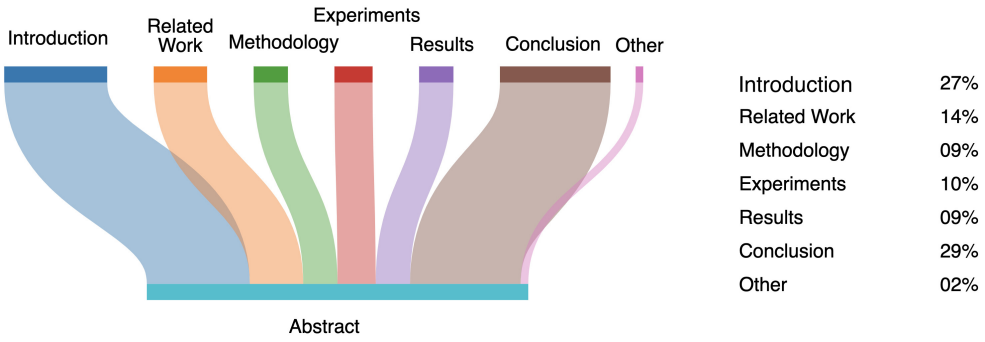

Fig. 4.7: Average involvement for the abstract with percentages of each section of research papers in our dataset

4.4 Relevance Matrix

The importance of section-wise partitioning becomes particularly evident when working with research papers, where each section serves a unique and specialized role. A research paper typically follows a structured format, including sections such as the introduction, literature review, methodology, results, discussion, and conclusion. Each of these sections carries specific information that is essential to fully understanding the paper's contributions. For instance, the introduction sets the stage for the research problem, while the literature review provides the necessary background and context by reviewing relevant studies. The methodology explains how the research was conducted, and the results present the findings. Finally, the discussion interprets those findings, and the conclusion sums up the research's contributions and potential future directions.

When summarizing a research paper, it becomes crucial to treat each of these sections distinctly, because they each contribute differently to the overall meaning of the paper. A summary that fails to take into account these differences will likely miss key points or misrepresent the research. Figure 4.7 highlights the contribution of various sections to the abstract. This visual representation emphasizes how some sections may provide more substantial input into the summary, particularly the abstract, than others. For example, the methodology and results sections often contain dense, highly technical information, while the discussion and conclusion are more interpretive and can contribute significantly to a well-rounded summary.

Moreover, different sections may also require different summarization weights. The breakdown of a paper into sections, as shown in Figure 4.7, thus not only contributes to a more accurate and comprehensive summary but also guides the choice of summarization method to improve the overall quality of the summary. This demonstrates the critical role that section-wise partitioning plays in research paper summarization, ensuring that the unique contributions of each section are adequately captured and represented.

Utilizing the relevance matrix proved to be an essential step in our summarization process, as it enabled us to assign precise priority percentages to the sentences extracted from each section of a research paper. This matrix functioned as a guiding framework that allowed us to quantify the importance of different sections relative to one another and ensure that the most critical information was given the appropriate emphasis in the final summary. The relevance matrix not only enhanced the precision of our extractive summarization but also ensured that the selected sentences carried the most valuable insights from each section of the paper.

In particular, we assigned the highest priority to the **Conclusion** and **Introduction** sections. The rationale behind this decision was based on the fact that these sections typically encapsulate the core message of the paper. The **Introduction** lays the groundwork for the research, presenting the problem, its significance, and the objectives of the study. It serves as a critical point of engagement, offering readers a clear understanding of the paper's purpose and scope. On the other hand, the **Conclusion** section distills the findings of the research, summarizing the key outcomes and their implications, and often includes suggestions for future work. Therefore, these two sections are particularly rich in content that directly influences the overall summary, and prioritizing them ensures that the essential elements of the paper are effectively captured.

To generate the relevance matrix, we relied on a dataset comprising 10,000 research papers. This dataset was particularly valuable because it provided comprehensive coverage of all six major sections commonly found in research papers: **Introduction**, **Related Work**, **Methodology**, **Experiments**, **Results**, and **Conclusion**. Each of these sections serves a distinct purpose in the overall structure of a research paper, and by having access to such a dataset, we were able to develop a nuanced understanding of how different sections contribute to the overall meaning of the paper. For instance, the **Related Work** section provides context and background, while the **Methodology** details the research approach. The **Experiments** and **Results** sections contain empirical data and findings, which are crucial but may require deeper interpretation to fit into a summary format.

The limited dataset of 10,000 papers, although constrained in size, was sufficient to identify patterns and assign appropriate priority levels to each section. This ensured that our relevance matrix was robust enough to handle a variety of paper types across different fields. However, it is worth noting that as the dataset expands, there is potential to further refine the relevance matrix, improving its accuracy and adaptability. Despite the dataset's limitations, our approach demonstrated that section-wise relevance and priority assignment could significantly enhance the quality of research paper summarization. By prioritizing the most critical sections, particularly the **Introduction** and **Conclusion**, we ensured that our summaries were not only concise but also representative of the key findings and contributions of the research.

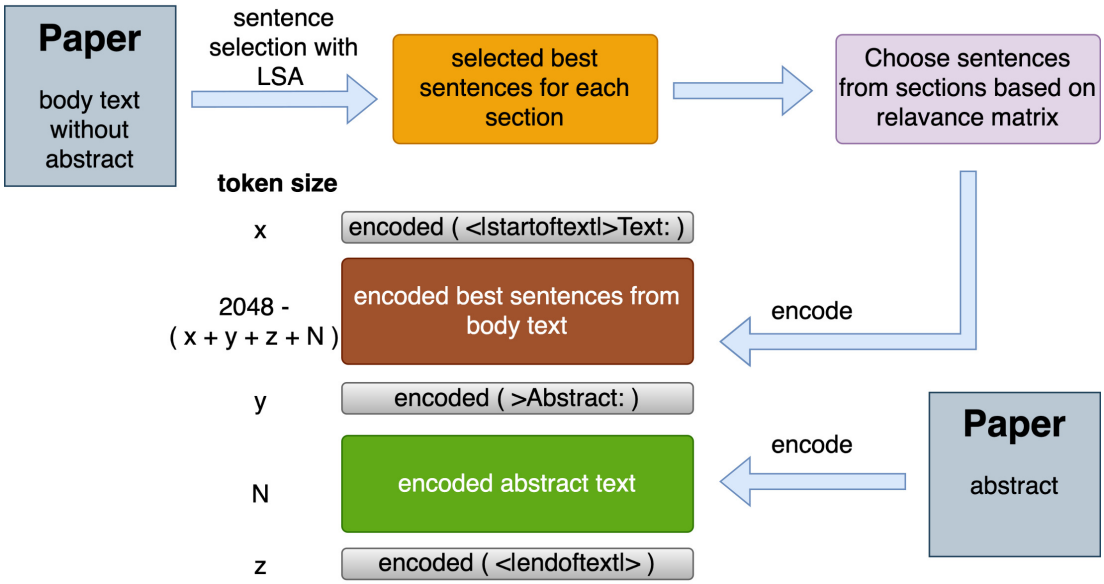

Fig. 4.8: Paper encoding

4.5 Encoding Data

The encoding process was a critical step in preparing the data for input into the GPT-Neo transformer model. After selecting the most relevant sentences from each section of the research papers based on the relevance matrix, the next phase involved ensuring that the data was properly formatted and tokenized to be compatible with the model. The relevance matrix was employed to prioritize the importance of sentences extracted from the various sections, ensuring that the most significant content from each section such as the introduction, methodology, results, and conclusion were selected for inclusion in the summarization process. This section-wise selection was necessary to maintain the structural integrity of the research paper and to generate coherent and contextually rich summaries.

Once the sentences were selected, they were processed using the GPT-2 tokenizer [56]. The GPT-2 tokenizer is particularly efficient for tasks involving large-scale text data as it breaks down the input into subword units, allowing for better handling of rare words and ensuring that the tokenized output is compact and optimized for transformer models. The tokenizer operates by converting the selected sentences into numerical representations, where each token corresponds to a subword or word fragment. This step was crucial for the subsequent stages, as transformer models like GPT-Neo require tokenized inputs to process and generate meaningful output.

After tokenizing the data, it was essential to store it in a format that allowed for efficient processing and retrieval, especially when dealing with large datasets. For this purpose, we chose to save the tokenized data as TFRecords, which is a TensorFlow-based data format designed for efficient reading and writing of large datasets. TFRecords not

only ensure that the data is compact and easy to handle but also optimize the speed of data access during training and inference. By using this format, we were able to minimize the input/output bottlenecks that often occur when working with large-scale machine learning models.

To further facilitate the accessibility and scalability of the data, we utilized Google Cloud[4] Storage as the cloud-based solution for storing and managing our TFRecord files. Google Cloud Storage offers several advantages for data-intensive applications, such as high availability, scalability, and integrated security features. The use of cloud storage allowed us to easily share and access the encoded data across different systems and computing environments, ensuring that our model training and experimentation process was not hindered by storage limitations or data accessibility issues.

The decision to use cloud storage was particularly beneficial for handling the large dataset of research papers, which contained thousands of documents and required robust infrastructure for storage and retrieval. Furthermore, Google Cloud's integration with TensorFlow made it easy to load the TFRecords directly into our model pipelines, ensuring seamless data processing. Additionally, the ability to scale storage capacity on demand was crucial for accommodating future expansions of the dataset or modifications to the data encoding process.

In summary, the data encoding process involved several key steps, including the use of the relevance matrix to reselect important sentences, tokenization with the GPT-2 tokenizer, and the storage of tokenized data as TFRecords on Google Cloud Storage. This combination of techniques ensured that our data was not only efficiently processed but also easily accessible and scalable, laying a solid foundation for the subsequent stages of model training and evaluation.

4.6 Model Tuning

After the dataset was prepared through the process of text size reduction and tokenization using the GPT-2 tokenizer, we proceeded to fine-tune the GPT-Neo model [59]. Fine-tuning is a crucial step in customizing a pre-trained model to better perform on specific tasks, in this case, generating summaries for research papers. The fine-tuning process involved adjusting the pre-trained weights of the GPT-Neo model using the tokenized dataset, allowing the model to learn the unique patterns and relationships present in the research paper summaries.

Fine-tuning was conducted using Google Colab[5], an accessible platform that provides free computing resources, including Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) . TPUs were chosen over traditional Central Processing Units (CPUs) or GPUs because they offer substantial advantages in terms of

[4] https://cloud.google.com
[5] https://colab.research.google.com/

both speed and computational efficiency for large-scale machine learning tasks. The use of TPUs significantly accelerated the training process, enabling us to handle the extensive dataset and the computational demands of fine-tuning GPT-Neo without sacrificing performance or increasing training time.

In addition to utilizing Google Colab's computing resources, both the dataset and the pre-trained GPT-Neo model were stored on Google Cloud. This setup facilitated seamless integration between storage and computing resources, allowing for quick loading of data during training and ensuring that large datasets were easily accessible without the need for local storage management. By using cloud-based infrastructure, we were able to scale up our experiments, adjusting batch sizes and training steps as needed without being constrained by hardware limitations.

The fine-tuning process involved several key configuration settings that were carefully selected to optimize the model's performance. A **batch size of 8** was chosen, which represents the number of training examples processed together in one iteration. While larger batch sizes typically enable faster convergence, the choice of a relatively moderate batch size helped maintain a balance between memory efficiency and model learning accuracy.

To further optimize the utilization of TPUs, we employed a **mesh shape configuration of x:4 and y:2**. This configuration determines how the TPUs are connected and organized in a grid-like structure to distribute the workload efficiently. By assigning four TPU cores along the x-axis and two along the y-axis, the mesh shape ensured that computations were distributed evenly across the TPUs, maximizing the hardware's computational capabilities and minimizing potential bottlenecks.

The model was trained for **1000 training steps**, with checkpoints saved every 500 steps. A training step represents one update of the model's weights based on the gradients calculated from the batch of data processed. This number of steps was chosen to provide a sufficient number of weight updates to fine-tune the model effectively without overfitting it to the training data. By saving checkpoints every 500 steps, we were able to monitor the model's progress and performance over time, enabling us to make adjustments if necessary. Checkpoints allowed us to capture intermediate versions of the model, ensuring that even if training were interrupted or needed to be paused, we could resume from the latest saved point.

Figure 4.9 illustrates the training process, showing key details such as the loss curve over time and the overall progression of model fine-tuning. Monitoring these metrics was essential in ensuring that the model was learning effectively and that the loss was decreasing as expected over the course of training. The fine-tuning process involved continuous adjustments and evaluations, and checkpoints were particularly useful for diagnosing any issues in model training, such as unexpected increases in loss, which could signal overfitting or improper learning.

By combining the computational power of TPUs, the cloud-based infrastructure of

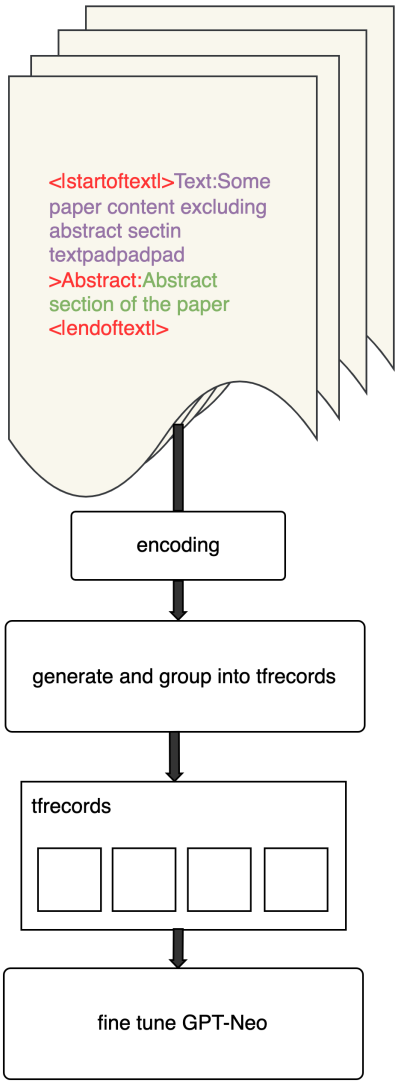

Fig. 4.9: Fine-tuning GPT-Neo

Google Cloud, and the fine-tuning capabilities of the GPT-Neo model, we were able to achieve a tailored model that is optimized for summarizing research papers. The configuration settings, batch size, mesh shape, training steps, and checkpoints played an integral role in balancing performance, efficiency, and accuracy, ensuring that the fine-tuning process yielded high-quality, contextually relevant summaries.

4.7 Prediction

After the GPT-Neo model [59] was fine-tuned, it was deployed to predict the summaries of research papers. This prediction phase involved generating abstract-level summaries from the encoded text extracted through the pre-summarization methods described earlier. The model, having been trained and fine-tuned on a specific dataset

of research papers, was now capable of taking input sequences (pre-summarized and tokenized texts) and producing coherent, contextually rich summaries that align with the structure and content of the original papers.

Similar to the fine-tuning process, predictions were conducted using Google Colab, leveraging **TPUs** for efficient and accelerated computation. The use of TPUs during the prediction phase allowed for faster inference, which is essential when dealing with large datasets or performing batch processing. In machine learning applications, inference refers to the process where the model uses its learned parameters (from the fine-tuning phase) to make predictions on new, unseen data. In this case, the model was tasked with generating summaries for various research papers based on the pre-summarized, tokenized input text.

To ensure consistency between the training and prediction phases, we maintained the same **batch size of 8** and **mesh shape configuration of x:4 and y:2** during the prediction phase. By keeping these parameters constant, we ensured that the model's performance remained stable and efficient. The batch size determines how many input texts were processed simultaneously during inference. A batch size of 8 meant that for each forward pass, eight research papers' pre-summarized texts were fed into the model, and eight summaries were generated as output. The mesh shape configuration, which distributed the computation across TPU cores in a 4x2 grid, optimized the workload and made the prediction process faster without compromising accuracy or computational resources.

The encoded text used in the prediction phase came from various sections of the research papers, selected through the pre-summarization methods. These methods, which prioritized specific sections such as the introduction and conclusion, ensured that the input to the GPT-Neo model was representative of the most important content in the paper. By feeding these encoded, tokenized texts into the fine-tuned model, we were able to generate high-quality summaries that retained the key points and overall meaning of the research papers.

The prediction process was not only efficient but also highly scalable. Since Google Colab provided the flexibility to handle large amounts of data via TPUs and cloud storage, we were able to predict summaries for multiple research papers simultaneously, significantly reducing the time required for inference. This scalability is particularly important when working with large datasets like those from arXiv, PubMed, or other section-divided datasets.

The predictions made by the fine-tuned GPT-Neo model were compared to the reference abstracts from the research papers, and the quality of these predictions was evaluated using various ROUGE metrics, as previously discussed. These metrics allowed us to quantify how well the generated summaries aligned with the original abstracts in terms of content coverage, sequence matching, and coherence. The use of a consistent batch size and mesh shape configuration during prediction ensured that the model pro-

duced summaries at a comparable level of quality across different research papers and datasets.

In conclusion, the prediction phase involved using the fine-tuned GPT-Neo model to generate summaries of encoded research paper texts, leveraging TPUs in Google Colab for efficient and scalable inference. By maintaining the same batch size and mesh shape configuration used during training, we ensured consistency in the model's performance, allowing it to generate accurate, contextually rich summaries based on pre-summarized and tokenized input texts. The combination of TPUs, cloud-based infrastructure, and careful pre-summarization enabled efficient and high-quality predictions across multiple datasets.

4.8 Training Flow

Key steps in our hybrid method are visually represented in Figure 4.10. We began by gathering an extensive collection of research papers from renowned sources, namely the **arXiv**, **PubMed** [39], and with our new dataset.

A crucial factor in our hybrid model was the need to make it computationally resource-efficient since both efficiency and availability should be considered. This involves identifying the hurdles presented by the very high computational costs associated with complex NLP models, and then, adopting a strategic approach by implementing Google's freely available Tensor Processing Units (TPUs)[6] for both training and predictions. For us to achieve high-performance computing at a low cost, the TPUs from Google which offer high throughput with low energy consumption seemed like the best solution. TPUs display an edge in regards to processing speed, with particular application to the matrix operations so important in deep learning models. Our case study confirms that it is possible to design and utilize prominent NLP models even under low-resource conditions.

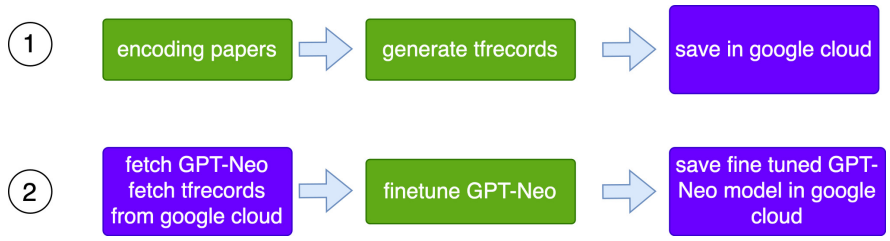

Fig. 4.10: Training flow. 1. Generation of tfrecords, 2. Fine-tuning GPT-Neo

The finetune was done as batch size of 8 and mesh shape of x:4, y:2 to match the computational resources. We used train steps of 1000 with steps per checkpoint was 500.

[6] https://cloud.google.com/tpu

CHAPTER 5

RESULTS

5.0.1 Pre-summarization methods comparison

Separately fine-tuned GPT-Neo models were evaluated for each pre-summarizer, as shown in Table 5.1 and Figure 5.1. It was observed that Latent Semantic Analysis (LSA) based pre-summarizations obtained the best results for the tested ROUGE scores.

TABLE 5.1: ROUGE SCORES COMPARISON OF THE MODELS BASED ON THE PRE-SUMMARIZATION METHOD

Pre-Summarization Method	ROUGE-1	ROUGE-2	ROUGE-L
Vector Average	0.2291	0.0250	0.1232
Lex Rank	0.3193	0.0636	0.1599
Text Rank	0.3196	0.0620	0.1601
LSA	**0.3213**	**0.0637**	**0.1602**

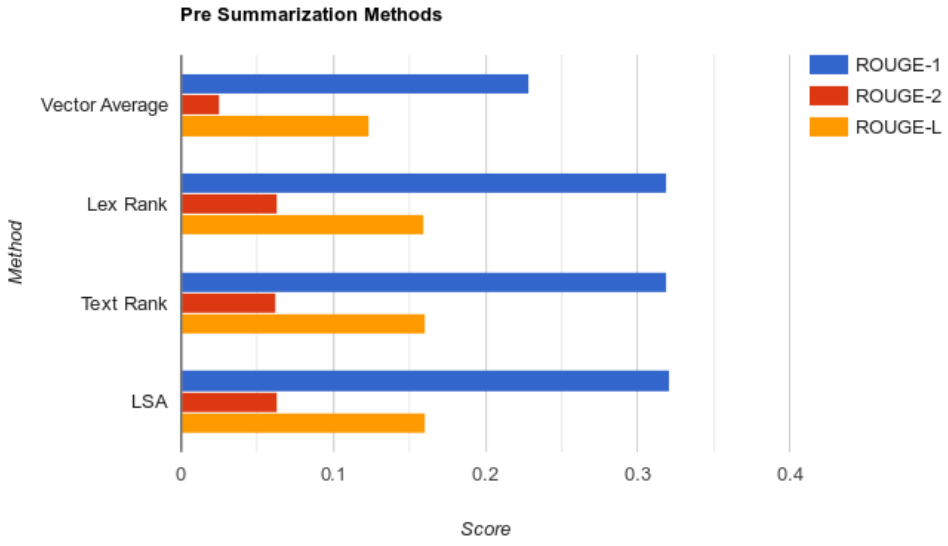

Fig. 5.1: ROUGE scores comparison of pre-summarization method

Further analysis was conducted on the configurations given in Table 5.1 by considering the Precision and Recall measures, as different research domains may prioritize one over the other. For example, de Silva [63] discuss how recall takes precedence over precision in medical domain NLP, and Samarawickrama et al. [64] mention similar trends in the legal domain. Although there is no definitive meta-study on the content

in computational linguistics research papers to conclude a bias towards precision or recall, it was deemed prudent to report these values. For ease of reading and comparison, the F1 values from Table 5.1 are also provided.

The average vector method takes the average of encoded vectors of text chunks before passing them into GPT-Neo for training or prediction. Although the average vector model may seem trivial for this task, recent work in the NLP domain has proven its usefulness in establishing a baseline for complex tasks such as sentiment analysis [65]. The results for this method are shown in Table 5.2.

TABLE 5.2: ROUGE SCORES OF AVERAGE VECTOR-BASED PRE-SUMMARIZING

ROUGE	F	P	R
1	0.2291	0.2675	0.2135
2	0.0250	0.0291	0.0234
L	0.1232	0.1452	0.1149

Lex Rank [61] is a stochastic graph-based method for computing the relative importance of textual units, based on the concept of eigenvector centrality in a graph representation of sentences. Similar, but mathematically simpler, methods have shown promise in NLP applications in the legal domain [66]. The model trained with Lex Rank provided the results shown in Table 5.3.

TABLE 5.3: ROUGE SCORES OF LEX RANK-BASED PRE-SUMMARIZING

ROUGE	F	P	R
1	0.3193	0.3561	0.3041
2	0.0636	0.0720	0.0602
L	0.1599	0.1785	0.1531

Since the advent of the PageRank algorithm [67, 68], using graph-based methods to rank text documents has become a popular solution for document-level analysis [69]. TextRank [60] is a graph-based sentence extraction method that creates a graph for each sentence and ranks them based on similarity. Sugathadasa et al. [70] showed how TextRank can be utilized in representing documents in a semantically consistent manner in their legal document retrieval system. Pre-summarization based on this method scored as shown in Table 5.4.

TABLE 5.4: ROUGE SCORES OF TEXT RANK-BASED PRE-SUMMARIZING

ROUGE	F	P	R
1	0.3196	0.3558	0.3039
2	0.0620	0.0697	0.0584
L	0.1601	0.1780	0.1527

LSA (Latent Semantic Analysis) [62] extracts and represents the contextual-usage meaning of words through statistical computations applied to the text. The ROUGE scores of this method as a pre-summarizer with GPT-Neo are shown in Table 5.5.

TABLE 5.5: ROUGE SCORES OF LSA-BASED PRE-SUMMARIZING

ROUGE	F	P	R
1	0.3213	0.3580	0.3070
2	0.0637	0.0710	0.0606
L	0.1602	0.1788	0.1534

LSA-based pre-summarization scored the highest on ROUGE-1, ROUGE-2 and ROUGE-L. All extractive pre-summarizations scored more than twice the score of the baseline, the vector average method, in ROUGE-2.

5.0.2 Evaluation with Datasets

The summarized version of the results in our research is shown in Table 5.6.

5.0.3 arXiv Dataset

As demonstrated in Table 5.6, applying a stemmer for text comparison led to a noteworthy achievement: the highest ROUGE-1 score, outperforming the results reported by Cohan et al. [39]. By incorporating a stemmer, we effectively reduced words to their base or root forms, enabling a more robust and accurate matching process. This preprocessing step addressed variations in word forms and inflections, ensuring that the comparison was based on semantic content rather than exact lexical matches.

As a result of this enhancement, our summarization technique achieved a ROUGE-1 score that surpassed the scores reported in the influential work by Cohan et al. [39]. This achievement demonstrates the effectiveness of our approach in generating more concise and contextually relevant summaries that closely align with the original reference text.

The comparison with the previous work by Cohan et al. [39] serves as an essential benchmark, showcasing the progress and advancements made in the field of text summarization. It validates the efficacy of our approach and positions our research as a noteworthy contribution to the domain.

5.0.4 PubMed Dataset

In addition to evaluating our model with the arXiv dataset, we further assessed its performance using the PubMed dataset, a crucial step to ensure the generalizability and robustness of our findings. The results of this comprehensive evaluation are presented

TABLE 5.6: ROUGE SCORE COMPARISON OF THE MODELS ON DATASETS. R1: ROUGE-1, R2: ROUGE-2, R3: ROUGE-3, RL: ROUGE-L

Pre-Summarization Method	R1	R2	R3	RL
arXiv Dataset				
Cohan et al. [39]	35.80	**11.05**	**3.62**	**31.80**
This work	33.77	8.73	2.67	18.57
This work + stemmer	**36.12**	9.56	2.96	19.55
PubMed Dataset				
Cohan et al. [39]	**38.93**	**15.37**	**9.97**	**35.21**
This work	33.47	8.88	3.39	18.28
This work + stemmer	35.85	9.64	3.66	19.06
Section-Divided New Dataset				
This work + relevance matrix	34.14	7.66	2.37	17.51
This work + relevance matrix + stemmer	**37.41**	8.51	2.60	18.66

in Table 5.6, highlighting the model's ability to handle a diverse range of scientific literature beyond the arXiv domain.

The outcome of this evaluation was pivotal in validating the versatility and performance of our model across different scientific domains. The results showcased its proficiency in handling the unique linguistic nuances and terminology present in medical research articles, effectively summarizing complex concepts and findings with a high level of accuracy.

5.0.5 New Dataset

As shown in Table 5.6, the highest ROUGE-1 value for arXiv data summaries was achieved by our model on the new dataset when a stemmer was applied. This result underscores the impact of the section-wise relevance matrix on boosting the model's performance.

Our primary research objective centered on exploring section-wise involvement-based summarization. The introduction of our novel dataset was crucial for this experimentation, allowing us to leverage the relevance matrix values as guiding metrics.

During the evaluation phase, we emphasized the priority sections within the paper that contributed substantially to the overall content. This strategic prioritization enabled us to focus on the most valuable and informative aspects of research papers, providing a more nuanced and contextually relevant approach to the summarization process. By honing in on the sections that matter the most, our research aimed to enhance the efficiency and effectiveness of summarization techniques.

CHAPTER 6

CONCLUSION

A key focus of our research was the assessment and improvement of research paper summarization, utilizing an innovative method that incorporates a section-wise relevance matrix. Expanding on this idea, we aimed to make a significant contribution to the field by creating a groundbreaking dataset of research papers, carefully segmented into sections, designed to address the unique requirements of the research community.

By adopting this approach, we demonstrated the importance of section-wise involvement, with a specific focus on the relevance of the *abstract* section. These involvement ratios were essential in refining and fine-tuning our model, particularly for extractive sentence selection. Our study introduces a novel and comprehensive methodology for research paper summarization, which has the potential to greatly improve the accessibility and clarity of research papers, benefiting both the academic and scientific communities.

To address the token size limitation, we implemented a two-part strategy. First, we utilized extractive summarization techniques to compress lengthy texts into concise, relevant sentences. By selecting key sentences through extractive summarization and refining them based on the **Relevance Matrix**, we successfully reduced the input size to meet the token constraints of the GPT-Neo model. The relevance matrix allowed us to prioritize the critical sections of the text, ensuring that essential information was preserved while maintaining the model's processing efficiency.

The second part of our method involved applying the GPT-Neo model for abstractive summarization, allowing it to generate summaries from the extracted sentences. Abstractive summarization builds on the extractive summary, producing new, coherent summaries that convey the core meaning of the original text. By using this hybrid approach, we leveraged the advantages of both extractive and abstractive techniques, resulting in detailed and informative summaries that effectively capture the main content of the research papers.

Our experiments and evaluations indicated that this combined approach produced promising outcomes, aligning closely with established benchmarks for summarization tasks. Despite the constraints imposed by limited token size, our method proved effective in generating high-quality summaries, highlighting the potential of transfer learning and language models in abstractive summarization.

It is worth noting that this study serves as a crucial foundation for enhancing the performance of this method. With further research and refinements, the approach has the potential to surpass existing benchmarks and set new standards in the field of summarization.

The practical implications of our findings are substantial. By overcoming the token

size limitation of LLMs using extractive summarization, we have created a pathway for more efficient text summarization across various fields. As high-quality language models become more accessible, our approach offers a promising solution for generating effective and insightful summaries of lengthy documents in a resource-efficient and cost-effective manner, benefiting researchers, educators, and information users alike.

Our method relies significantly on the section-wise relevance matrix to balance the summary bias. However, this approach can face limitations in cases where the division of sections is not clearly defined. In such scenarios, accurately determining and applying the involvement ratios can be difficult, which may reduce the accuracy of the summary. Additionally, as noted by Nenkova and McKeown [71], the Latent Semantic Analysis (LSA) technique often requires multiple sentences to capture all relevant information, presenting another potential limitation in our pre-summarization process.

In conclusion, this study highlights the potential of transfer learning with LLMs, specifically GPT-Neo, for text summarization tasks. By leveraging GPT-Neo's language generation abilities and the adaptability of LSA's semantic analysis techniques combined with our relevance matrix, we improved both the performance and accuracy of our approach. We addressed the token size limitation by using extractive summarization to condense lengthy documents, followed by abstractive summarization to create comprehensive summaries. Incorporating section-wise involvement further enhanced summarization quality. The promising results emphasize the need for continued research to refine and expand this method. Ultimately, this advancement has the potential to transform the summarization field in low-resource environments, facilitating the extraction of key insights from large text sources and supporting more efficient decision-making.

REFERENCES

[1] D. Kumarasinghe and N. De Silva, "Automatic generation of abstracts for research papers," in *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*, 2022, pp. 221–229.

[2] D. Kumarasinghe and N. de Silva, "Abstract generation with hybrid model supported by relevance matrix," in *International Conference on Computational Collective Intelligence*. Springer, 2024, pp. 211–223.

[3] L. Huang, Y. He, F. Wei, and W. Li, "Modeling document summarization as multi-objective optimization," 04 2010, pp. 382–386.

[4] Resoomer, "Summarizer to make an automatic text summary online." [Online]. Available: https://resoomer.com/en

[5] SummarizeBot, "Get to know more by reading less!" [Online]. Available: https://www.summarizebot.com/

[6] SMMRY, "Summarize articles, text, websites, essays and documents." [Online]. Available: https://smmry.com/

[7] TLDRThis, "Tldr this." [Online]. Available: https://www.tldrthis.com/

[8] TextCompactor, "Text compactor." [Online]. Available: https://www.textcompactor.com/

[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[10] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[11] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[13] N. Nazari and M. Mahdavi, "A survey on automatic text summarization," *Journal of AI and Data Mining*, vol. 7, no. 1, pp. 121–135, 2019. [Online]. Available: http://jad.shahroodut.ac.ir/article_1189_162.html

[14] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113679, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420305030

[15] N. Moratanch and C. Gopalan, "A survey on extractive text summarization," 01 2017, pp. 1–6. [Online]. Available: https://www.researchgate.net/publication/317420253_A_survey_on_extractive_text_summarization

[16] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: a brief survey," *arXiv preprint arXiv:1707.02268*, 2017.

[17] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, 01 2017. [Online]. Available: https://www.researchgate.net/publication/299499824_Recent_automatic_text_summarization_techniques_a_survey

[18] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," in *2016 International Conference on Circuit, power and computing technologies (ICCPCT)*. IEEE, 2016, pp. 1–7.

[19] K. S. Jones *et al.*, "Automatic summarizing: factors and directions," *Advances in automatic text summarization*, pp. 1–12, 1999.

[20] E. Hovy, C.-Y. Lin *et al.*, "Automated text summarization in summarist," *Advances in automatic text summarization*, vol. 14, pp. 81–94, 1999.

[21] S. Dutta, V. Chandra, K. Mehra, S. Ghatak, A. Das, and S. Ghosh, *Summarizing Microblogs During Emergency Events: A Comparison of Extractive Summarization Algorithms: Proceedings of IEMIS 2018, Volume 2*, 01 2019, pp. 859–872.

[22] A. Mahajani, V. Pandya, I. Maria, and D. Sharma, "A comprehensive survey on extractive and abstractive techniques for text summarization," *Ambient Communications and Computer Systems*, pp. 339–351, 2019.

[23] "D u c 2 0 0 7 p a s t & d a t a." [Online]. Available: https://www-nlpir.nist.gov/projects/duc/data/2007_data.html

[24] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[25] L. Yang, X. Cai, Y. Zhang, and P. Shi, "Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization," *Information sciences*, vol. 260, pp. 37–50, 2014.

[26] R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova, and C. A. Mehdiyev, "Mcmr: Maximum coverage and minimum redundant text summarization model," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14 514–14 522, 2011.

[27] J.-g. Yao, X. Wan, and J. Xiao, "Compressive document summarization via sparse optimization," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[28] ——, "Phrase-based compressive cross-language summarization," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 118–127.

[29] J. Zhang, X. Cheng, G. Wu, and H. Xu, "Adasum: an adaptive model for summarization," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 901–910.

[30] Y. Ouyang, W. Li, S. Li, and Q. Lu, "Applying regression models to query-focused multi-document summarization," *Information Processing & Management*, vol. 47, no. 2, pp. 227–237, 2011.

[31] C. Li, Y. Liu, and L. Zhao, "Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 778–787.

[32] P. Li, L. Bing, W. Lam, H. Li, and Y. Liao, "Reader-aware multi-document summarization via sparse coding," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[33] W. Li, F. Wei, Q. Lu, and Y. He, "Pnr2: Ranking sentences with positive and negative reinforcement for query-oriented update summarization," in *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)*, 2008, pp. 489–496.

[34] H. Liu, H. Yu, and Z.-H. Deng, "Multi-document summarization based on two-level sparse representation model," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[35] H. Jin, Y. Zhang, D. Meng, J. Wang, and J. Tan, "A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods," *arXiv preprint arXiv:2403.02901*, 2024.

[36] "Github - abisee/cnn-dailymail: Code to obtain the cnn / daily mail dataset (non-anonymized) for summarization," 2021. [Online]. Available: https://github.com/abisee/cnn-dailymail

[37] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," *arXiv preprint arXiv:1808.08745*, 2018.

[38] A. R. Fabbri, I. Li, T. She, S. Li, and D. R. Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," *arXiv preprint arXiv:1906.01749*, 2019.

[39] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," *arXiv preprint arXiv:1804.05685*, 2018.

[40] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," *arXiv preprint arXiv:1810.09305*, 2018.

[41] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345*, 2019.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[43] "The new york times annotated corpus." [Online]. Available: https://catalog.ldc.upenn.edu/LDC2008T19

[44] "Byte cup 2018 international machine learning contest." [Online]. Available: https://www.biendata.xyz/competition/bytecup2018/

[45] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[46] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[47] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[48] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[49] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in neural information processing systems*, vol. 28, 2015.

[50] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[51] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural abstractive text summarization with sequence-to-sequence models," *ACM/IMS Trans. Data Sci.*, vol. 2, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3419106

[52] M. Grusky, "Cornell newsroom summarization dataset," 2021. [Online]. Available: https://lil.nlp.cornell.edu/newsroom/

[53] "Github - tshi04/nats: Neural abstractive text summarization with sequence-to-sequence models," 2021. [Online]. Available: https://github.com/tshi04/NATS

[54] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[55] G. Moro, N. Piscaglia, L. Ragazzi, and P. Italiani, "Multi-language transfer learning for low-resource legal case summarization," *Artificial Intelligence and Law*, vol. 32, no. 4, pp. 1111–1139, 2024.

[56] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[57] M. Ravaut, A. Sun, N. Chen, and S. Joty, "On context utilization in summarization with large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 2764–2781.

[58] J. Kreutzer, I. Caswell, L. Wang, A. Wahab, D. van Esch, N. Ulzii-Orshikh, A. Tapo, N. Subramani, A. Sokolov, C. Sikasote *et al.*, "Quality at a Glance: An

Audit of Web-Crawled Multilingual Datasets," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 50–72, 2022.

[59] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, "GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow," Mar. 2021, If you use this software, please cite it using these metadata. [Online]. Available: https://doi.org/10.5281/zenodo.5297715

[60] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[61] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[62] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[63] N. H. N. D. de Silva, "Semantic oppositeness for inconsistency and disagreement detection in natural language," Ph.D. dissertation, University of Oregon, 2020. [Online]. Available: https://search.proquest.com/openview/3def92003aaab0901d896bf3919034ba/1?pq-origsite=gscholar&cbl=18750&diss=y

[64] C. Samarawickrama, M. de Almeida, N. de Silva, G. Ratnayaka, and A. S. Perera, "Party identification of legal documents using co-reference resolution and named entity recognition," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2020, pp. 494–499.

[65] V. Jayawickrama, G. Weeraprameshwara, N. de Silva, and Y. Wijeratne, "Seeking sinhala sentiment: Predicting facebook reactions of sinhala posts," *arXiv preprint arXiv:2112.00468*, 2021.

[66] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, and B. Ayesha, "Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings," in *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*. IEEE, 2017, pp. 79–84.

[67] L. Page, "Method for node ranking in a linked database," *USA Patent*, vol. 6, 1997.

[68] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[69] E. L. Karannagoda, H. M. T. C. Herath, K. N. J. Fernando, M. W. I. D. Karunarathne, N. H. N. D. de Silva, and A. S. Perera, "Document Analysis Based Automatic Concept Map Generation for Enterprises," in *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*. IEEE, December 2013, pp. 154–159.

[70] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal, and M. Perera, "Legal Document Retrieval using Document Vector Embeddings and Deep Learning," in *Science and information conference*. Springer, 2018, pp. 160–175.

[71] A. Nenkova and K. McKeown, "A survey of text summarization techniques," *Mining text data*, pp. 43–76, 2012.