Large Language Models for Effective and Efficient Text Summarization

HARISH NANJUNDAPPA

Mid Thesis Report

APRIL 2024

DEDICATION

ACKNOWLEDGEMENTS

ABSTRACT

News text summarization is the task of producing a short and accurate summary of a news article that captures the main information and highlights. It is a useful application for information retrieval, content analysis, and news aggregation. However, news text summarization poses several challenges, such as dealing with complex and diverse topics, preserving factual accuracy, and generating fluent and coherent summaries. In recent days, Large Language Models (LLMS) have demonstrated a great deal of promise for improving summarization methods. Our goal in this work is to investigate competing LLMS for text summarization, including BERT, T5 and GPT-4, from the perspectives of architecture, pre-training, fine-tuning, and assessment. The CNN/DailyMail news dataset will be used for this work, and the performance of the models will be evaluated against metrics like ROUGE and BLEU. This study highlights the pros and cons of each strategy, along with the remaining challenges in LLM-based text summarization. The aim is to offer insights into the effectiveness of particular language models in text summarization and to stimulate new ideas and research in the field.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABS: Abstractive Text Summarization

AI: Artificial Intelligence

ATS: Automatic Text Summarizer

BiLSTM: Bi Directional Long Short Term Memory Network

BERT: Bidirectional Encoder Representations from Transformers

BLEU: Bi-Lingual Evaluation Understudy

CNN: Convolutional Neural Network

DL: Deep Learning

ETS: Extractive Text Summarization

GLUE: General Language Understanding Evaluation

GPT: Generative Pre-trained Transformer

GPU: Graphics Processing Unit

GRU: Gated Recurring Unit

LLM: Large Language Model

LM: Language Model

LSTM: Long Short-Term Memory Network

MDS: Multi Document Summarization

ML: Machine Learning

MLM: Masked Language Model

MMLU: Massive Multitask Language Understanding

NLG: Natural Language Generation

NLP: Natural Language Processing

OOV: Out Of Vocabulary

PLM: Pre-trained Language Model

RNN: Recurrent Neural Network

ROUGE: Recall-Oriented Understudy for Gisting Evaluation

SDS: Single Document Summarization

T5: Text-to-Text Transfer Transformer

TF-IDF: Term Frequency Inverse Document Frequency

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

News Text Summarization is the task of producing a concise and relevant summary of a news article. This task can help readers get the main information quickly, as well as assist journalists, editors and researchers in analyzing large amounts of news data. Text summarization techniques and models can be broadly classified into two categories: extractive and abstractive. Extractive methods select salient sentences or phrases from the input text and concatenate them to form a summary, On the other hand, sentences in source texts are rewritten using abstractive methods, which is more in line with how humans tackle the same issue. In theory, abstractive methods might produce summaries that seem more natural and are more efficient than extractive methods

The evolution of text summarization has seen significant progress since Hans Peter Luhn's pioneering work in the 1950s (Luhn H P, 1958). Luhn's approach involved selecting sentences based on statistical properties, but it was limited by simple heuristics and an inability to capture semantic relationships. Traditional methods, including graph-based and machine learning (ML) approaches, emerged over subsequent decades but struggled with linguistic complexity.

The introduction of deep learning (DL), notably with neural network architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), revolutionized text summarization. DL models, such as sequence-to-sequence models (Sutskever et al., 2014) with attention mechanisms (Vaswani et al., 2017a), showed superior performance in generating abstractive summaries by learning from raw text data and attending to relevant information.

Large Language Models (LLMs), like BERT (Devlin et al., 2019), T5 (Raffel et al., 2019), GPT (Radford et al., 2019; Brown et al., 2020), have recently emerged as powerful tools for text summarization. Pre-trained on large text corpora, LLMs demonstrate remarkable capabilities in understanding and generating natural language text. Fine-tuning LLMs on summarization-specific objectives and prompts has led to state-of-the-art results in abstractive summarization across various domains. Prompt learning and few-shot learning techniques further enhance LLM-based summarization by enabling models to adapt quickly to new tasks and domains with

minimal training data. These methods leverage carefully designed prompts and examples to guide the model's generation process, facilitating efficient transfer learning and adaptation.

Despite the significant progress made in LLM-based summarization, several challenges and pending issues remain. These include the generation of informative and coherent summaries under constraints such as length and diversity, the handling of factual accuracy and bias, and the development of evaluation metrics that accurately assess the quality of generated summaries. Addressing these challenges will be crucial for further advancing the field of text summarization and unlocking its full potential in real-world applications.

This thesis aims to assess the effectiveness and limitations of LLMs for news text summarization while proposing methods to enhance their performance. It involves a thorough literature review of LLMs and news summarization techniques, comparing different LLMs and summarization approaches. Experiments will evaluate summary quality and diversity using automatic metrics, examining factors like input/output length, domain, and style. Novel methods like domain adaptation, style transfer, prompt engineering will be proposed and tested. Ethical and social implications of LLM usage will be discussed alongside future research directions.

1.2 Problem Statement

Text summarization is a critical task in NLP aimed at condensing lengthy documents into concise and informative summaries. With the advent of LLMs, such as BERT, T5, and GPT, there is a growing interest in leveraging these models for text summarization tasks. However, while these models have demonstrated impressive capabilities across various NLP tasks, their suitability and comparative performance in text summarization remain to be thoroughly evaluated.

The study focuses on evaluating the performance of BERT, T5, and GPT in generating high-quality summaries. It will delve into their respective strengths and weaknesses, exploring how fine-tuning and prompting strategies impact their effectiveness. By conducting a comparative analysis, the research seeks to provide insights into the relative advantages and trade-offs of each model.

The findings of this study will contribute to advancing the understanding of LLMs in text summarization and provide valuable insights into the strengths and limitations of each model. Furthermore, the comparative analysis will inform researchers and practitioners in selecting the most suitable LLM for specific summarization tasks, thereby facilitating the development of more effective and efficient text summarization systems.

1.3 Research Questions

The following queries are attempted to be addressed by this study:

1. How does the performance of BERT, T5, and GPT compare in terms of generating high-quality summaries on news articles?
2. What are the specific strengths and weaknesses of each LLM in terms of summarization quality and computational efficiency?
3. How do different fine-tuning strategies, such as domain adaptation and prompt engineering, impact the summarization performance of each LLM?
4. How do architectural differences between BERT, T5, and GPT affect their performance in text summarization tasks?
5. How do different evaluation metrics, beyond ROUGE scores, capture the nuances of summarization quality provided by each LLM?
6. How can the findings from this comparative analysis inform the development of more effective and efficient text summarization systems using LLMs?

1.4 Aim & Objectives

The aim of this study is to conduct a comprehensive comparative analysis of LLMs for text summarization, specifically focusing on BERT, T5, and GPT models.

Research Objectives:

1. Conduct an exhaustive analysis of the existing literature about the LLM based text summarization on news datasets.

2. Implement data preprocessing tasks like removal of special characters, tokenization, sentence splitting, padding, truncation and many other required data preparation steps.
3. Evaluate the effectiveness of BERT, T5, and GPT in generating high-quality summaries.
4. Assess the strengths and weaknesses of each LLM in terms of summarization quality, coherence, and computational efficiency.
5. Investigate the impact of fine-tuning strategies, training data size, and task-specific parameters on the summarization performance of each LLM.
6. Provide insights into the comparative advantages and trade-offs of BERT, T5, and GPT in text summarization tasks, enabling informed decision-making for researchers and practitioners.

1.5 Scope of the Study

The scope of research for the comparative analysis of BERT, T5, and GPT-4 in news text summarization encompasses several key aspects:

- Data Pre-Processing: Study will address the data pre-process steps required to train, fine tune and evaluate the LLMs on text summarization task.
- Model Comparison: The study will compare the performance of BERT, T5, and GPT-4 in generating summaries from news articles.
- Fine-Tuning Techniques: Investigation of various fine-tuning strategies for adapting the pre-trained language models to the task of news text summarization. This may include different learning rates and optimization algorithms.
- Prompting Strategies: Evaluation of different prompting techniques on the summarization quality for GPT-4 model. This may involve experimenting with few shots learning and separate prompts for extractive and abstractive text summarization.
- Evaluation Metrics: Utilization of few evaluation metrics to assess the quality of the generated summaries. This includes standard metrics like ROUGE and BLEU.
- Interpretation of Results: Provide an in-depth analysis of the findings, including insights into the strengths and weaknesses of each model, factors influencing their performance, and potential areas for improvement.

- Practical Implications: Discuss the practical implications of the research findings for real-world applications, such as automated news summarization systems and content recommendation engines.
- Limitations and Future Directions: Identify limitations of the study and propose directions for future research to address these limitations and further advance the state-of-the-art in news text summarization using language models.

By delineating these aspects within the scope of the research, the study aims to offer a comprehensive understanding of how BERT, T5, and GPT-4 perform in the context of news text summarization and provide valuable insights for researchers and practitioners in the field.

Reasons for selection of BERT, T5, GPT-4 for this research are highlighted below:

- State-of-the-Art (SOTA): BERT, T5, and GPT-4 are among the most advanced and widely used LLMs in NLP. They have demonstrated SOTA performance across various NLP tasks, including text summarization.
- Versatility: BERT excels in capturing bidirectional context, making it suitable for understanding relationships within the text. T5 adopts a text-to-text approach, allowing it to handle diverse NLP tasks as text generation problems. GPT-4 builds on the success of its predecessors in generating coherent and contextually relevant text.
- Diverse Architectures: BERT, T5, and GPT-4 employ different architectures and pre-training objectives, leading to diverse representations and capabilities. By comparing these models, the study can provide insights into how different architectural choices and pre-training objectives impact their performance in text summarization tasks.
- Research Interest and Relevance: Given the widespread use and interest in these models within the NLP research community, comparing these LLMs for text summarization can contribute towards advancing our understanding of their strengths, weaknesses, and suitability for different applications.
- Availability of Pre-trained Models: Simplifies fine tuning and evaluation of text summarization, without the need for extensive computational resources or data.
- Potential for Improvement: Despite their success, there may still be room for improvement in text summarization tasks. By comparing these LLMs, the study can

identify areas where each model excels and potential avenues for further research and model development.

1.6 Significance of the Study

The landscape of text summarization has typically been dominated by approaches based on statistical, machine learning, and deep learning paradigms, which have been extensively researched and refined throughout time. In contrast, the LLM-based method represents a relatively new but quickly emerging frontier in the field, with continuous research activities and growing attention among scientists. Despite its promising potential, the LLM-based approach remains relatively underexplored in comparison to its more established counterparts, warranting further investigation and exploration. Through the contribution of code, benchmarks, and research findings, this endeavor seeks to narrow this gap, enriching the existing body of literature with valuable insights and advancements in LLM-based text summarization methodologies. In addition to its focus on LLM-based methods, this study also investigates prompt and few short learnings-based text summarizing approaches, offering light on innovative strategies and methodologies employed in this developing field of study.

In terms of application, this work will help journalists and editors to quickly and accurately summarize large amounts of information from various sources, such as press releases, reports, interviews, and social media. This can save time and resources, as well as enhance the readability and relevance of the news articles. News publishers can focus on generating summaries for different audiences, purposes, and platforms, such as headlines, abstracts, bullet points, tweets, or newsletters

1.7 Structure of the Study

The structure of the study is as follows:

Chapter 1 – Introduction: This chapter introduces the research problem and establishes the context for the subsequent chapters. It outlines the significance of this research and sets the stage for reader to understand the motivation behind this study.

Chapter 2 – Literature Review: This chapter covers existing research work on text summarization and identifies any gaps that serve as the foundation for the current study.

Chapter 3 – Research Methodology: This chapter describes the approach and procedures used to conduct the research, including algorithms, techniques, dataset, evaluation metrics and tools. This chapter serves as a guide for readers to understand how the study was conducted and how the results are analyzed.

Chapter 4 – Implementation: This chapter details the technical implementation and different experiments followed in the current study. It illustrates how theoretical concepts were converted into workable solutions by highlighting the inventiveness and ingenuity required in the research process.

Chapter 5 – Results & Evaluation: This chapter covers the results of experiments conducted during study and provides the empirical evidence through visualization and metrics to support the study claims and hypothesis.

Chapter 6 – Conclusion & Future Work: This chapter summarizes the main findings of the research and recommends directions for future research. It highlights any limitations or challenges encountered during the study. Finally, this chapter provides closure to the research narrative and highlights the contribution of this study.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter discusses the previous works done in the field of text summarization.

2.2 Overview of Text Summarization

2.2.1 Classification of Text Summarization Systems

Different methods are utilized to create text summarization systems, as depicted in Figure 2.1. Brief introduction of these methods is covered in the following section.

Classification by input size of documents: Single Document Summarization (SDS) or Multi Document Summarization (MDS). In SDS, single document is used to generate text summary (Garner, 1982), whereas in MDS approach, summaries are produced based on set of input documents by removing the redundant and repetitive information (Ferreira et al., 2014) .

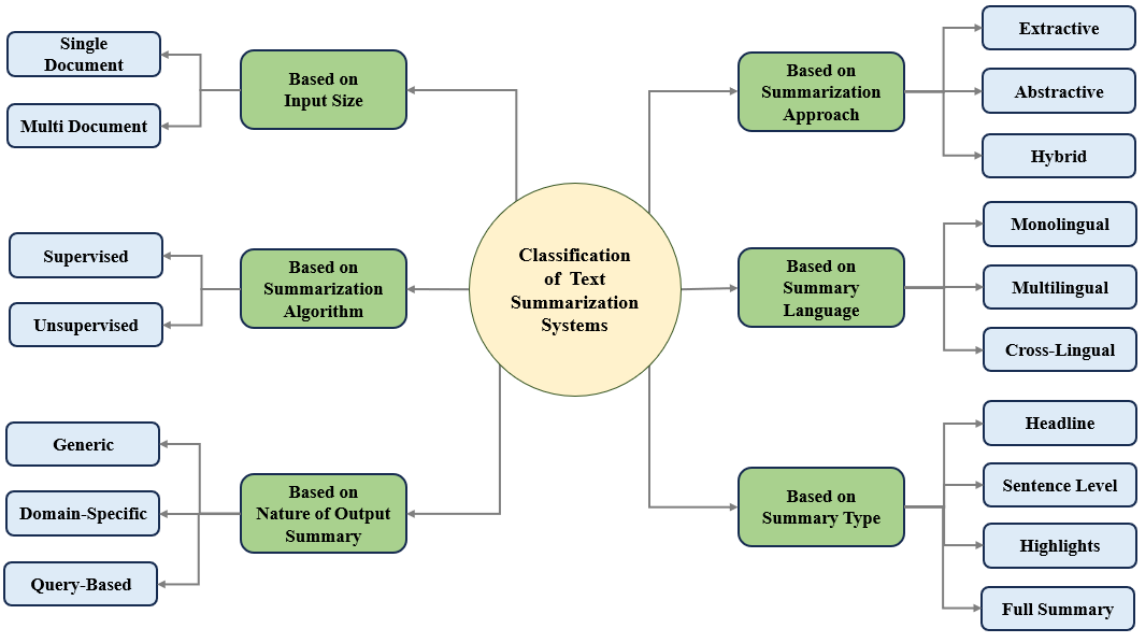

Figure 2.1: Classification of Text Summarization Systems

References - (El-Kassas et al., 2021; Radhakrishnan and Senthil kumar, 2023)

Classification by summarization approach: Summarizers can be extractive, abstractive or hybrid. In extractive text summarization (ETS) key sentences or phrases are used to generate summaries (Luhn H P, 1958; Edmundson, 1969; Rau et al., 1989). Abstractive text summarization (ATS) is complex and involves paraphrasing the input text content. All these approaches involving rewriting techniques like sentence compression (Knight and Marcu, 2000; Cohn and Lapata, 2009) sentence fusion (Barzilay and Mckeown, 2005), sentence revision (Tanaka et al., 2009a) and sentence splitting (Genest and Lapalme, 2012a). Hybrid method includes combination of both abstractive and extractive approaches (Hovy and Lin, 1996).

Classification based on algorithms: Supervised or unsupervised depending on the model learning technique. For supervised approach, annotated data (labeled data) or human summaries will be required to train the language model (Mandal et al., 2021), where as in unsupervised approach (Barzilay and Lee, 2003; Lakshmi and Latha, 2022; Zhao et al., 2022; Kurisinkel and Chen, 2023)model learns from unlabeled data by identifying patterns and relationships in the data. Supervised approach requires manual annotation of training data which involves lot of human effort and thus makes challenging and costly to train the summarizer systems.

Classification by summary languages: Summarizers can be monolingual, multilingual, or cross-lingual. In monolingual, input and output content are of similar language (Kutlu et al., 2010; Lukas, 2021; Prithwiraj Bhattacharjee, 2021) , where as in multilingual scenario the input content can be in multiple languages (e.g., English or Kannada) and the output summary will also be generated in these languages (Hovy and Lin, 1996; Foroutan et al., 2022; Alcantara et al., 2023; Taunk and Varma, 2023). Cross-lingual summarizer takes input content in one language (e.g., English) and output summaries in different languages (e.g., Kannada, Hindi) (Shree Akshaya et al., 2022; Zheng et al., 2022; Takeshita et al., 2023).

Classification by nature of output summaries: Summarizers can be of generic, query-based or domain specific summarizers. Aim of generic text summarizer is to obtain abstract summary from input documents. Query-based summarizer also known as ‘Topic-based’ or ’User focused’ leverages the query information from user input to generate summaries as per user needs (Ramakrishna et al, 2006; Hennig Leonhard, 2009; Tang Jie et al., 2009; Gambhir and Gupta, 2017). Domain specific summarizers excel in summarizing for specific field (e.g., legal, medical), offering high relevance and accuracy (Ghosh et al., 2022; Zakraoui et al., 2022; Qin

and Luo, 2023; Shaik et al., 2023), while general text summarizers (Ballout et al., 2023) are more versatile and applicable across multiple domains.

Classification by summary type: Depending on the type of summary desired, summarizers can be implemented to generate headlines, highlights, or full summaries. Headlines provide concise titles that capture the essence of a longer text, making it easier for readers to quickly grasp the main idea (Banko et al., n.d.; Sepúlveda-Torres et al., 2021; Singh et al., 2021). Highlights offer key points or snippets from the text, giving readers an overview of the most important information. Full summaries provide comprehensive synopses of the entire text, condensing its contents into a shorter form while retaining the essential details. Each type of summary serves a different purpose and can be tailored to meet specific needs, whether for skimming through content, extracting key insights, or understanding the main arguments.

2.2.2 Traditional Methods for Text Summarization

2.2.2.1 SDS and MDS

Single Document Summarization (SDS) or Multi Document Summarization (MDS). In SDS, single document is used to generate text summary (Garner, 1982), whereas in MDS approach, summaries are produced based on set of input documents by removing the redundant and repetitive information (Ferreira et al., 2014) . Compared to SDS, MDS is more complex to implement as it involves redundancy, temporal dependency, compression, and many more challenges. Maximal Marginal Relevance (MMR) technique was presented which aimed to reduce the redundancy issue (Carbonell and Goldstein, 1998). Still redundancy remains a challenge. In the study conducted by (Li et al., 2020), Neural abstractive MDS model was proposed that can analyse multiple input documents and generate abstractive summaries more efficiently by utilizing popular graph representations of text. Graph based MDS approach proposed by (Gunes Erkan, 2011) highlight the limitations of extractive MDS summarization, with limited coverage, and difficulty in handling redundancy, which can impact the informativeness and quality of the summary.

2.2.2.2 Extractive Summarization

Extractive methods have been dominant in text summarization research for a long time, due to their simplicity and effectiveness. Extractive text summarization has progressed from the earliest statistical techniques to machine learning strategies, deep learning models, and finally, large language models (LLMs). Some of the common methods of extractive text is depicted in Figure 2.2 and highlighted in below section.

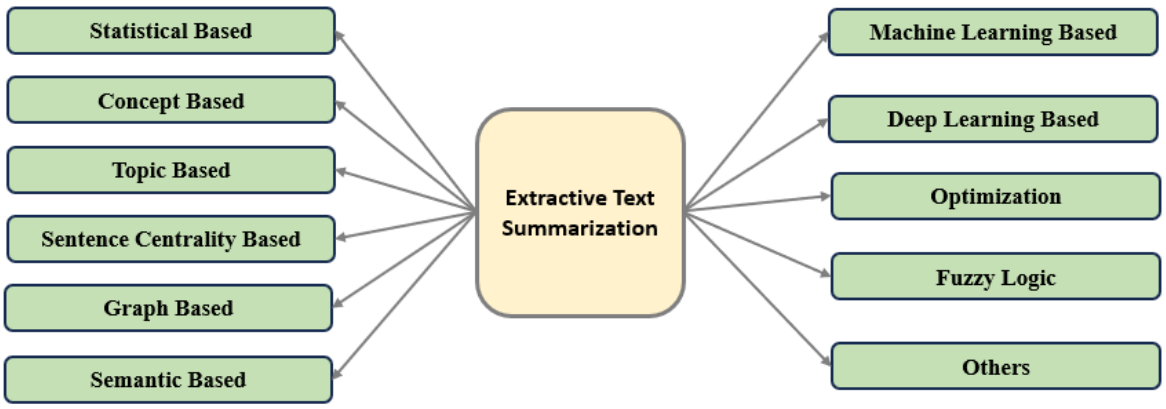

Figure 2.2: Extractive Summarization Methods

References - (El-Kassas et al., 2021)

Frequency-based methods: These methods use the frequency of words or n-grams as a measure of their importance and select the sentences or phrases that contain the most frequent words or n-grams. Examples of frequency-based methods are Luhn's method (Luhn H P, 1958), TF-IDF (Salton and Buckley, 1988), and Edmundson's method (Edmundson, 1969).

Position-based methods: These methods use the position of sentences or phrases in the original text as a measure of their importance and select the sentences or phrases that appear in the beginning, end, or specific sections of the text. Examples of position-based methods are Lead, Last, and First-sentence methods. (Brandow et al., 1995; Mani and Maybury, 1999)

Cue-based methods: These methods use the presence of cue words or phrases in the original text as a measure of their importance and select the sentences or phrases that contain the cue words or phrases. Examples of cue words or phrases are titles, headings, keywords, or indicators of topic shifts or conclusions. Examples of cue-based methods are Hovy and Lin's method (Hovy and Lin, 1996) and Kupiec et al.'s method (Kupiec et al., 1995)

Graph-based methods: These methods use a graph representation of the original text, where the nodes are sentences or phrases and the edges are the similarity or relatedness between them, and apply graph algorithms to rank and select the sentences or phrases. Examples of graph-based methods are TextRank (Mihalcea and Tarau, 2004), LexRank (Gunes Erkan, 2011), and Submodular methods (Lin and Bilmes, 2010)

Machine learning methods: Both supervised and unsupervised approaches are being used to classify the text document. Supervised learning approach need human annotated (labelled data) summaries to train the model during which the model willl learn the features and weights that determine the salience and relevance of the sentences or phrases, and use them to rank and select the sentences or phrases. Some of the summarization study involving supervised learning approaches are Support Vector Machine (Fattah, 2014a; Mandal et al., 2021), Random Forest (Ansamma John and M Wilscy), KNN(Mandal et al., 2021), Naïve Bayes classification (Fattah, 2014a), Mathematical Regression (Fattah and Ren, 2009), Decision trees (Mandal et al., 2021), Multilayer Perceptron (Fattah and Ren, 2009).

Unsupervised systems do not require training data and generate summary by accessing only target documents, attempting to discover hidden structures. They use heuristic rules to extract relevant sentences and generate summaries. Examples include clustering (Yang et al., 2014), K-Means clustering and Hidden Markov Models (Pascale Fung and Chi-Shun, 2003). Genetic algorithms (GA) are an evolutionary algorithm that solve optimization problems using natural evolution approaches like mutation, inheritance, crossover, and selection (Fattah and Ren, 2009; Mendoza et al., 2014; Kumar Meena and Gopalani, 2015; Chen et al., 2021; Tomer and Kumar, 2022) .

Deep learning & LLMs: These methods use neural network models, such as Recurrent Neural Networks (RNNs) (Hochreiter S and Jurgen S, 1997; Dey and Salem, 2017) Convolutional Neural Networks (CNNs) (O'Shea and Nash, 2015) or attention mechanisms (Vaswani et al., 2017a), to learn the semantic and syntactic representations of the sentences or phrases, and use them to rank and select the sentences or phrases. Examples of deep learning methods are RNN-Ext (Cheng and Lapata, 2016), CNN-Ext (Chen and Bansal, 2018), and BERT (Liu and Lapata, 2019a).

Fuzzy Logic: This method uses fuzzy logic to assess sentence importance in a document, considering linguistic variables like word frequency and semantic relevance. This approach allows for degrees of truth and adaptability to linguistic uncertainty, resulting in concise summaries that effectively capture key points (Azhari and Kumar, 2017; Goularte et al., 2019; Patel et al., 2019; Du and Huo, 2020).

Research gaps & limitations:

- Lack of readability and coherence – Summaries generated by this approach tends to be disjointed and ungrammatical by ignoring the linguistic and logical aspects between the sentences or phrases (Barzilay and Lapata, 2008; Li et al., 2019).
- Extractive summarization methods often lack diversity & novelty – This approach tends to produce summaries that are repetitive, redundant, and lexically similar to original text. (Carbonell and Goldstein, 1998)
- Lack of user preference or feedback – This approach tend to produce fixed or static summaries without considering the preference, need or feedback of users
- Lack of generalization and domain adaptation – This approach performs well on specific domain and fails to adapt or generalise for other domain or genres.

2.2.2.3 Abstractive Summarization

Different approaches for abstractive summarization are depicted in Figure 2.3. Broadly they are classified into three categories: structure-based, semantic-based and deep learning approaches.

Structure Based ATS:

Structured-based abstractive text summarization is a method that condenses text while preserving key information and context. It uses tree-based, template-based, ontology-based, lead and body phrase methods, rule-based, and graph-based approaches to identify key phrases, relationships, and domain-specific ontologies. These methods ensure coherence, relevance, and adaptability to various languages and domains. Some of the research works in this field are: tree based (Barzilay and Mckeown, 2005), template-based method (Harabagiu Sanda and Finley Lacatusu, 2001) , rule based (Ansamma John and M Wilscy; Genest and

Lapalme, 2012b; Le and Le, 2013) , ontology based (Lee et al., 2005), lead and body phrase (Tanaka et al., 2009b), graph based (Ganesan et al., 2010; Wang and Li, 2012).

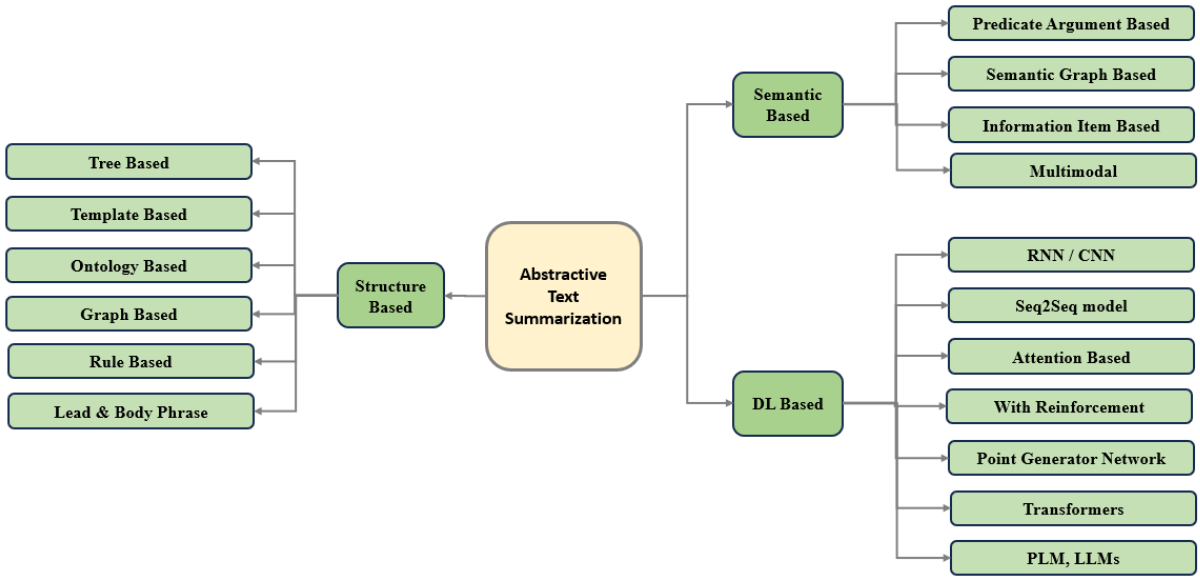

Figure 2. 3: Abstractive Text Summarization Methods

Structure-based approaches enhance coherence, content selection, summary structure, and domain adaptability by considering input text's structural organization. However, they face challenges in representing text structural elements, handling real-world text variability, and limiting output diversity compared to more flexible, semantic methods.

Semantic-based:

The methods represent input documents using semantic representations like information items, predicate-argument structures, or semantic graphs, which are then fed to a natural language generation (NLG) system for the final abstractive summary. A few research works on these methods are multimodal semantic model (Gatt and Reiter, 2009), information item-based method (Genest and Lapalme, 2011), semantic graph model (Leskovec et al., 2004; Ibrahim et al., 2012; Lloret et al., 2015; Munot and S. Govilkar, 2015), semantic text representation model (Khan Atif and Naomie Salim, 2015; Khan et al., 2015)

Advantages of this method includes improved coherence and adaptability, and disadvantages with this approach is it adds computational complexity and interpretability challenges.

Deep Learning approaches:

The recent breakthroughs in deep learning and neural networks have enabled significant progress in abstractive text summarization. Neural abstractive methods use sequence-to-sequence models, which consist of an encoder that encodes the input text into a vector representation, and a decoder that generates the summary from the vector representation, using attention mechanisms to concentrate on relevant parts of the input text (Sutskever et al., 2014; Chorowski and Bahdanau, 2015). Neural abstractive methods can generate fluent and readable summaries with less human effort and domain knowledge and can also incorporate copy or pointer mechanisms to deal with out-of-vocabulary words or rare entities (Rush et al., 2015; Chopra et al., 2016; Paulus et al., 2017; See et al., 2017; Gū et al., 2019).

Challenges:

Despite the advances in neural abstractive methods, they still face several challenges, such as factual consistency, content selection, and evaluation. Neural abstractive methods can sometimes generate summaries that contain factual errors or inconsistencies with the input text, due to the lack of explicit reasoning or verification mechanisms (Cao et al., 2018; and Falke et al., 2019). Neural abstractive methods can also struggle with selecting the most salient and relevant information from the input text, especially when the input text is long or complex, or when the summarization goal is specific or query-focused (Nallapati et al., 2016b; Gehrmann et al., 2018). Moreover, neural abstractive methods are still difficult to evaluate, as the existing automatic metrics, such as ROUGE (Lin, 2004) or BLEU (Papineni et al., 2002), do not capture the semantic or pragmatic aspects of summarization quality, and the human evaluation is costly and subjective (Papineni et al., 2002; Lin, 2004; Liu and Lapata, 2019b).

2.2.2.4 Hybrid Summarization

Hybrid Automatic Text Summarization is a method that combines extractive and abstractive techniques to produce high-quality summary information. It involves selecting, merging, compressing, or deleting important information to obtain new summary information. Current research aims to match machine-generated abstracts to human-written ones. The typical architecture of a hybrid text summarizer consists of pre-processing, sentence extraction, summary generation, and post-processing. Two main methods used in hybrid text summarization are a) extractive to abstractive, b) extractive to shallow abstractive method.

Extractive to Abstractive - The methods begin with extractive methods and then apply abstractive summarization methods to the extracted sentences (Wang et al., 2017). Some of the work related to this approach are (Zeng et al., 2016; Gehrmann et al., 2018; Sahba et al., 2018; Rudra et al., 2019;)

Extractive to Shallow Abstractive - The methods begin with extractive ATS methods and then employ a shallow abstractive text summarization method, utilizing techniques such as information compression, information fusion (Lloret et al., 2013) and synonym replacement (Annapurna P Patil et al., 2014) to extract sentences (Sahoo et al., 2018).

Up until 1990, summarization experiments were conducted with an emphasis on extracting summaries from original text instead of abstracting newly generated content(Hovy and Lin, 1998). Later Semantic and statistical features were used to for extracting and abstracting the summaries. (Fattah, 2014b; Bhat et al., 2018; Alami et al., 2021)

With the increase in popularity of ML models, statistical techniques were combined with ML approaches (both supervised and unsupervised learning) to generate high quality summaries. Some of the studies based on ML approaches are TF with LEAD (Ishikawa et al., n.d.), TF with SVM(Gupta and Kaur, 2016), TF-IDF with SumBasic (Sharifi et al., 2014), TF-IDF with K-Means (Pandya, n.d.; Khan et al., 2019; Padmapriya and Duraiswamy, 2020), TF-IDF with DL and AE (Yousefi-Azar and Hamey, 2017).

Fuzzy approaches were used in combination of statistical approaches, for example, TF-IDF with fuzzy logic (Patel et al., 2019), TF-IDF with fuzzy logic and Apriori (Malallah et al., n.d.), TF-IDF, LSA with Fuzzy (Güran et al., 2017) , TF with fuzzy logic (Goularte et al., 2019).

Studies related to Genetic and metaheuristic algorithms were used to extract salient sentences which were later used to generate concise abstracts (Binwahlan et al., 2010; Al-Radaideh and Bataineh, 2018), Coot Remora Optimization (CRO) technique based on Deep RNN (Bandari and Bulusu, 2023), mayfly-harmony search with Multi Hidden RNN (Zeyad and Biradar, 2023)

In another work by (Gupta and Gupta, 2018)they proposed a hybrid approach using graph-based (PageRank), MMR, clustering (K-Means), fuzzy logic, feature-based extraction

technique. Their experiments showed hybrid approach gives either better results or comparable results than the individual techniques.

Recent advancement in Deep learning approaches involving CNN, RNN, Seq2Seq, Transformers, Attention mechanism, has significally improved the accuracy of text summarization. Some of the research in this field includes ANN based (Chintan Shah and Dr. Anjali Jivani, 2018) , CNN with Bi-LSTM and Multi-layer Perceptron (Abhishek Kumar Singh et al., 2019), CNN with LSTM and Capsule Networks (Cho et al., n.d.) , Two stage framework Bi-LSTM with LSTM and attention (Amplayo and Lapata, 2021)

Lately, Transformer based LLMs and PLMs have achieved State-of-the-work result for text summarization. Some of the examples include BART with T5 pre trained models (A. Ghadimi), TF-IDF and Transformers (Sophie and Siva Sathya, 2022), Pretrained T5 (Bishop et al., 2022), TF-IDF with T5 (Mugi Karanja and Matheka, 2022)

Finally, Hybrid text summarization has been successfully applied for different domain which includes Legal (Galgani et al., 2012; Anand and Wagh, 2022), BioMedical (Lloret et al., 2013), and other areas.

Challenges:
Balancing extractive and abstractive techniques are a significant challenge in summarization. Extractive methods produce coherent summaries but may miss important information, while abstractive methods generate comprehensive summaries but may introduce errors. Handling multi-document summarization adds complexity, and achieving human-like summaries requires incorporating natural language understanding and generation capabilities to resonate with human readers. These challenges must be addressed in future research to ensure effective summarization.

2.2.3 Role of LLMs in Advancing Text Summarization

Traditional approaches to text summarization involve extractive methods, which select and concatenate important sentences or phrases from the original text. However, these methods have limitations such as the inability to capture the overall meaning and coherence of the text, failure to tailor summaries to specific user needs, and difficulty in handling complex texts.

In contrast, recent advancements in text summarization leverage Large Language Models (LLMs) and Pre-trained Language Models (PLMs), which are neural network models trained on vast amounts of text data. LLMs and PLMs offer several advantages over traditional approaches:

- Abstractive Summarization: They can generate summaries that go beyond the original text by paraphrasing, rephrasing, or synthesizing new information that is relevant and concise.
- Feature Extraction: LLMs and PLMs excel at extracting meaningful features from text data, identifying important keywords, phrases, and sentences.
- Customization: These models can generate summaries tailored to different user preferences, including length, style, tone, or perspective.
- Handling Complex Texts: They are adept at handling diverse and complex texts, including multimodal, multilingual, or domain-specific content.
- Interactive Summarization: These models can be integrated into interactive summarization systems, enabling personalized summaries tailored to specific needs or preferences.
- Multi-Document Summarization: They can process multiple documents simultaneously, suitable for multi-document summarization tasks.
- Transfer Learning: PLMs leverage transfer learning, capturing general language patterns and knowledge from pre-training phases, beneficial for summarization tasks with limited training data.
- Evaluation and Metrics: LLMs and PLMs aid in developing better evaluation metrics for text summarization, enabling researchers to create robust metrics that accurately assess the quality and relevance of summaries.

In conclusion, LLMs and PLMs represent superior approaches to text summarization compared to traditional methods, enhancing the effectiveness and performance of summarization systems.

2.3 Previous Work on LLMs for Text Summarization

The use of large language models (LLMs) and pre-trained language models (PLMs) based on the Transformer architecture (Vaswani et al., 2017a) is one of the most recent developments in text summarization research. LLMs depends on large amounts of unlabeled text data to learn general language representations and generate natural language outputs BERT model (Devlin et al., 2019) Open GPT model (Radford et al., 2019), GPT 3 FSL (Brown et al., 2020).

(Paulus et al., 2017) proposed a novel approach that blends PLMs with reinforcement learning and graph neural networks to generate abstractive summaries that are more informative and consistent with the input text. RoBERTa (Liu et al., 2019) introduced a new pre-training objective for LLMs that encourages the model to generate concise and fluent summaries from long documents, without relying on any labeled summarization data.

LLMs can improve the performance and robustness of neural abstractive methods, as they can capture more semantic and syntactic information from the input text and generate more diverse and coherent summaries BERTSum (Liu and Lapata., 2019), PEGASUS (Zhang et al., 2020). LLMs can also enable zero-shot or few-shot learning for text summarization, where the model can generalize to new domains or tasks without fine-tuning or with minimal supervision UNILM(Dong et al., 2019) , T5, (Raffel et al., 2019) , BART (Lewis et al., 2019), BLOOM (Workshop et al., 2022), PaLM (Chowdhery et al., 2022), LaMDA(Thoppilan et al., 2022), LLaMA (Touvron et al., 2023)

Google has released a new family of multimodal models Gemini (Gemini Team et al., 2023) that show impressive text, audio, video, and image interpreting skills. This model is among the first to attain Human Performance on 30 out of 32 state-of-the-art benchmarks. Figure 2.4 and Table 2.1 illustrates the chronological timeline showcasing the development and progression of Large Language Models (LLMs) over time.

Challenges:
However, LLMs are not without limitations and challenges. LLMs require a large number of computational resources and memory to train and run, which poses ethical and environmental concerns and limits their accessibility and reproducibility (Schwartz et al., 2019, Strubell et al., 2019). LLMs can also suffer from factual inconsistency, content selection, and evaluation

issues, as they are not explicitly trained for text summarization and may not align with the summarization objectives or expectations (Kryściński et al., 2019; Fabbri et al., 2020; Goyal and Durrett., 2020). Moreover, LLMs can generate summaries that are biased, misleading, or harmful, due to the potential biases or noises in the pre-training data or the generation process (Gehman et al., 2020; Bender et al., 2021). Finally, challenges exist in effectively controlling the length, style, and tone of the generated summaries, and adaptation of the model to different summarization scenarios and user preferences. Therefore, text summarization research based on LLMs is still an emerging and promising direction, with many research questions and challenges to be addressed.

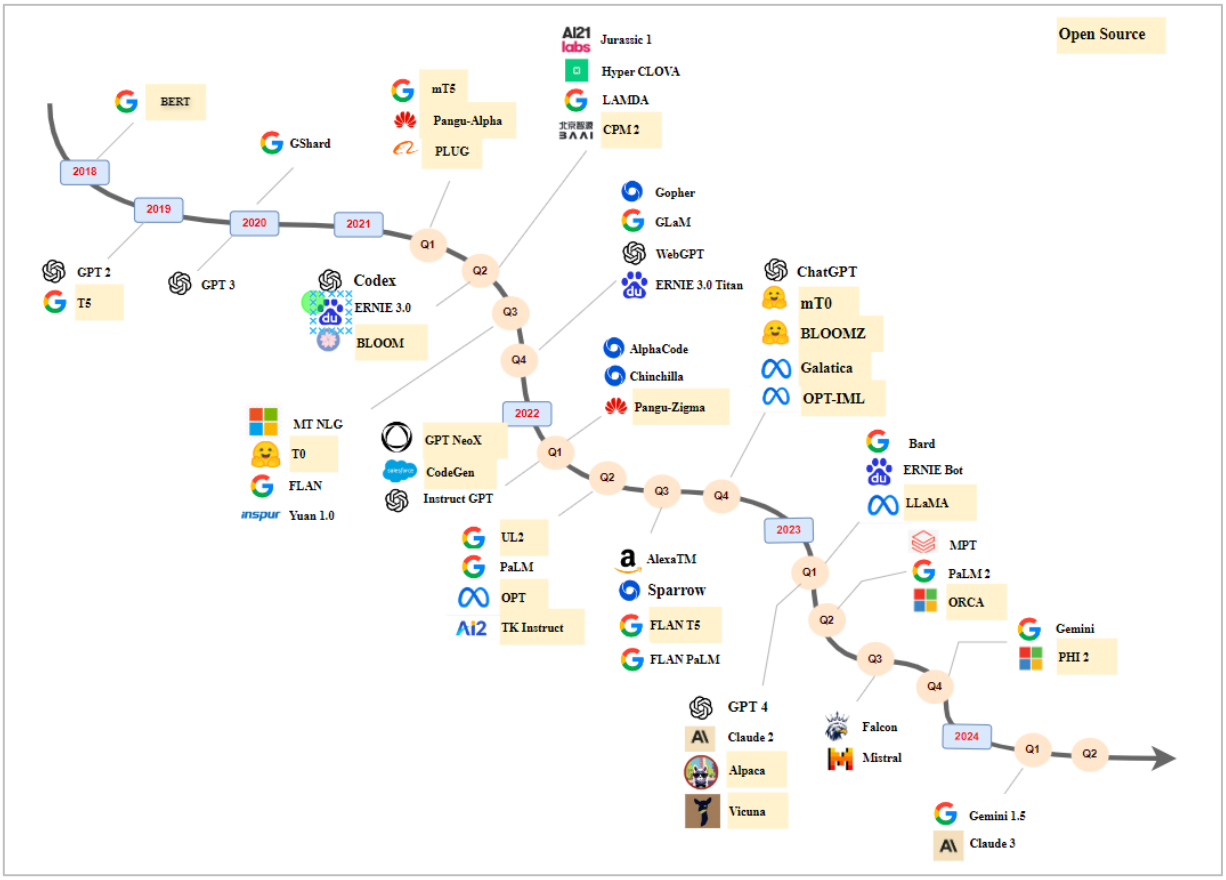

Figure 2.4: Evolutionary Paths of LLMs

References - (Zhao et al., 2023a)

The emergence of transformers in 2017 has triggered notable progress in large language models (LLMs) such as BERT, GPT, and T5, leading to a substantial transformation across various natural language processing (NLP) tasks. Figure 2.4 and Table 2.1 illustrates the evolutionary trajectory of LLMs over time.

Table 2. 1: Development of LLMs Over Time

Time Line	LLM	Parameters Size	Organization
2018 (Oct-Dec)	BERT	Base = 110 M, Large = 340 M	Google
2019 (Oct – Dec)	GPT 2	1.5 B	Open AI
2019 (Oct – Dec)	T5	T5 Small = 60 M T5 Base = 220 M T5 Large = 770 M T5 3B = 3 B T5 11B = 11 B	Google
2020 (Apr – Jun)	GPT-3	175 B	Open AI
2020	GShard	600 B	Google
2021 (Jan – Apr)	mT5	580 M	Google
2021 (Jan – Apr)	PanGu-Alpha	200 B	Huawei
2021 (Jan – Apr)	PLUG		Alibaba
2021 (May – Aug)	ERNIE 3	10 B	Baidu
2021 (May –Aug)	BLOOM	176 B	BigScience
2021 (May – Aug)	Codex	12 B	Open AI
2021 (May – Aug)	Jurassic 1	178 B	AI21 Labs
2021 (May – Aug)	HyperCLOVA	175 B	NAVER Labs
2021 (May – Aug)	LaMDA	137 B	Google
2021 (May – Aug)	CPM-2	11 B	BAAI
2021 (Sep – Oct)	GPT 3.5	175 B	Open AI
2021 (Sep – Oct)	FLAN	Flan-T5-Small = 80 M Flan-T5-Base = 250 M Flan-T5-Large = 780 M Flan-T5-XL= 3 B Flan-T5-XXL = 11 B	Google
2021 (Sep – Oct)	Yuan 1.0	245.7 B	Inspur
2021 (Sep – Oct)	MT-NLG	530 B	Microsoft
2021 (Sep – Oct)	T0	11 B	HuggingFace
2021 (Nov-Dec)	GLaM	1.2 T	Google
2021 (Nov-Dec)	Gopher	280 B	DeepMind(Google)
2021 (Nov-Dec)	WebGPT	1.76 T	Open AI
2021 (Nov-Dec)	Ernie 3.0 Titan	260 B	Baidu
2022 (Jan-Mar)	InstructGPT	1.76 T	OpenAI
2022 (Jan-Mar)	GPT-NeoX	20 B	EleutherAI
2022 (Jan-Mar)	CodeGen	16.1 B	Salesforce
2022 (Jan-Mar)	AlphaCode	41.4 B	DeepMind (Google)
2022 (Jan-Mar)	Chinchilla	1.4 T	DeepMind (Google)
2022 (Jan-Mar)	PanGu-Epsilon	1 T	Huawei
2022 (Apr-Jun)	TK_Instruct	NA	Ai2
2022 (Apr- Jun)	OPT	125 B to 175 B	Meta

2022 (Apr- Jun)	UL2	19.5 B	Google
2022 (Apr- Jun)	PaLM	540 B	Google
2022 (Jul- Oct)	GLM	NA	Zhipu AI
2022 (Jul- Oct)	AlexaTM	11 B	Amazon
2022 (Jul- Oct)	Sparrow	70 B	DeepMind (Google)
2022 (Jul- Oct)	Flan-PaLM	540 B	Google
2022 (Nov- Dec)	mT0	NA	HuggingFace
2022 (Nov- Dec)	BLOOMZ	3 B	HuggingFace
2022 (Nov- Dec)	Galatica	5 different variants, 125 M to 120 B	Meta
2022 (Nov- Dec)	OPT-IML	175B	Meta
2022 (Nov- Dec)	ChatGPT	1.5 B	OpenAI
2023 (Jan-Mar)	Aya	13 B	Cohere
2023 (Jan-Mar)	Bard	137 B	Google
2023 (Jan-Mar)	ERNIE Bot	10 T	Baidu
2023 (Jan-Mar)	LLaMA	65 B	Meta
2023 (Jan-Mar)	GPT-4	1.7 T	Open AI
2023 (Jan-Mar)	Claude 2	130 B	Anthropic
2023(Jan-Mar)	Vicuna	7B to 65B	LMSYS
2023(Jan-Mar)	Alpaca	7B to 65B	Stanford
2023(Apr-Jun)	Dolly 2.0	2.8 B to 12B	Databricks
2023(Apr-Jun)	MPT	30 B	Databricks
2023 (Apr-Jun)	PaLM 2	540 B	Google
2023 (Apr-Jun)	ORCA	7B	Microsoft
2023 (Jul-Sep)	OpenLLaMA	NA	Meta
2023 (Jul-Sep)	Falcon	7B	Technology Innovation Institute
2023 (Jul-Sep)	Mistral	7B	Mistral AI
2023 (Oct-Dec)	Gemini	18B	Google
2023 (Oct-Dec)	Phi 2	2.7 B	Microsoft
2024 (Jan-Mar)	Gemini 1.5	560B to 600 B	Google
2024 (Jan-Mar)	Claude 3	500 B	Anthropic

2.3.1 Evolution of LLMs/PLMs from Traditional Language Models

Natural Language Processing (NLP) faces the challenge of modelling the structure and dependencies of natural language, including syntax, semantics, and discourse. Recurrent neural networks (RNNs) like LSTMs and GRUs are commonly used for language modelling, which helps generate natural and fluent texts and pre-train models for downstream tasks. However, RNNs have limitations like parallelization difficulties and the vanishing or exploding gradient problem. Researchers propose attention mechanisms to overcome these limitations, focusing on relevant input and output sequences, encoding and decoding

efficiently, and capturing long-range dependencies and global information without sequential processing.

Natural language processing was revolutionized by (Vaswani et al., 2017a) when they introduced the Transformer architecture, which has a self-attention mechanism that allows it to recognize word dependencies regardless of where they are in a sequence. Tasks like language modelling, text summarization, and machine translation are well suited for the Transformer due to its encoder and decoder stacks, which consist of feed-forward layers and multi-head self-attention layers. The Transformer accelerates training and convergence and achieves state-of-the-art performance on a range of NLP tasks by utilizing residual connections, layer normalization, and masked self-attention during decoding.

2.3.2 Emergence of BERT, T5, GPT Series

Using Transformer as the fundamental architecture, researchers have improved models such as BERT, T5, and GPT, achieving state-of-the-art results and establishing new benchmarks for NLP research and applications.

The Transformer encoder serves as the foundation for BERT, a pre-trained language model that was presented by (Devlin et al., 2019). With the help of next sentence prediction and masked language modelling, BERT was trained on large text corpora to acquire bidirectional representations of words and sentences that include both syntactic and semantic information. BERT outperforms other downstream NLP tasks such as question answering, sentiment analysis, and natural language inference by fine-tuning with task-specific layers.

Using the encoder-decoder structure of the Transformer architecture, T5, a Text-To-Text Transfer Transformer, was proposed by (Raffel et al., 2019; Xue et al., 2020). T5 is pre-trained on a large text corpus with a single goal in mind: text-to-text denoising autoencoding. This allows it to reconstruct original sequences from corrupted inputs, including token insertion, deletion, and permutation. By framing these as text-to-text problems and fine-tuning input-output formats, this approach allows T5 to acquire flexible and robust representations of

text, suitable for a variety of tasks such as machine translation, text summarization, and classification.

GPT, or Generative Pre-trained Transformer, developed by (Radford et al., 2019) over several iterations, utilizes the decoder component of the Transformer architecture.GPT learns to predict the next token given the previous context by pre-training on large-scale text datasets with an emphasis on causal language modelling, excluding future tokens. Because of its ability to produce coherent and fluid text, it can be easily adjusted to a variety of downstream tasks, such as text generation, summarization, and classification. This is accomplished by adding linear layers to the decoder's output or fine-tuning it with task-specific modifications.

2.3.3 Adaptation of Language Models for Text Summarization

Training, pretraining, and fine-tuning are essential steps in adapting large language models (LLMs) for text summarization. Training provides a foundational understanding of language, pretraining enriches the model with general linguistic knowledge, and fine-tuning tailors the model to excel in summarization tasks, collectively enabling LLMs to comprehend input texts and generate accurate summaries effectively.

2.3.3.1 Pretraining LLMs for Text Summarization

The process of pre-training large language models (LLMs) for text summarization involves initially training them on vast, unlabeled datasets such as Wikipedia or news articles, followed by fine-tuning them for particular summarization tasks. This method allows LLMs to gain both broad and domain-specific knowledge, improving their ability to create accurate and relevant summaries. Through pre-training, LLMs can develop skills in extracting important information from source texts, structuring it logically, and generating coherent and natural language sentences, thereby enhancing their performance in summarization tasks.

Advantages gained by Pretraining LLMs for Text Summarization includes:

- Enhances generalization: Pretraining on diverse text corpora captures broad linguistic patterns and knowledge.

- Improves domain-specific knowledge: Fine-tuning on specific summarization tasks improves relevance and accuracy of summaries.
- Facilitates efficient learning: Pretraining provides a starting point for faster convergence and reduced data requirements.
- Improves information extraction: LLMs learn to extract important information from source texts, enabling effective key point identification and irrelevant detail discard.
- Improves coherence and fluency: Pretraining aids in organizing information logically, resulting in fluent and natural-sounding summaries.
- Reduces overfitting: Initializing the model with pretrained weights reduces risk of overfitting, resulting in robust performance on unseen data.

2.3.3.2 Fine-tuning Models for Text Summarization

Pre-trained LLMs for text summarizing can be improved by training them again on labeled datasets comprising source texts and summaries, which allows the model to be tailored to certain summarization applications. LLMs can gain task-specific expertise through this process, which improves their ability to produce pertinent and accurate summaries that are suited to various genres and specifications. LLMs perform better overall on text summarization tasks when they fine-tune their output, which includes learning to add domain-specific vocabulary, modify summary length, tone, and style, and ensure coherence and consistency in the created summaries.

Different fine-tuning approaches for pre-trained language models (LLMs) in text summarization include full fine-tuning, layer-wise fine-tuning, gradual fine-tuning, prompt-based fine-tuning, multi-task fine-tuning and knowledge distillation. These approaches vary in the extent to which they update model parameters and the strategies used to adapt the LLM to specific summarization tasks, considering factors like available data, computational resources, and desired model behavior

2.3.4 Prompt Engineering

Prompt engineering enables the enhancement of large language models (LLMs) by utilizing task-specific instructions, known as prompts, to guide model behavior without modifying core

parameters. Rather than updating model parameters, prompts facilitate seamless integration of pre-trained models into downstream tasks, providing contextual guidance or activating relevant knowledge through natural language instructions or learned vector representations. This technique enhances model adaptability and efficacy, empowering practitioners to tailor outputs efficiently to specific tasks and domains. Initial research and popularization of the concept of prompt engineering was done in the LLMs (Liu et al., 2021; Chen et al., 2023; Tonmoy et al., 2024).

2.3.3.2 Prompt Engineering Strategies and Techniques

This section offers a succinct summary of the development of prompting techniques, from zero-shot prompting to the most recent developments, and has arranged prompt engineering strategies based on their application areas as highlighted in Fig 2.4.

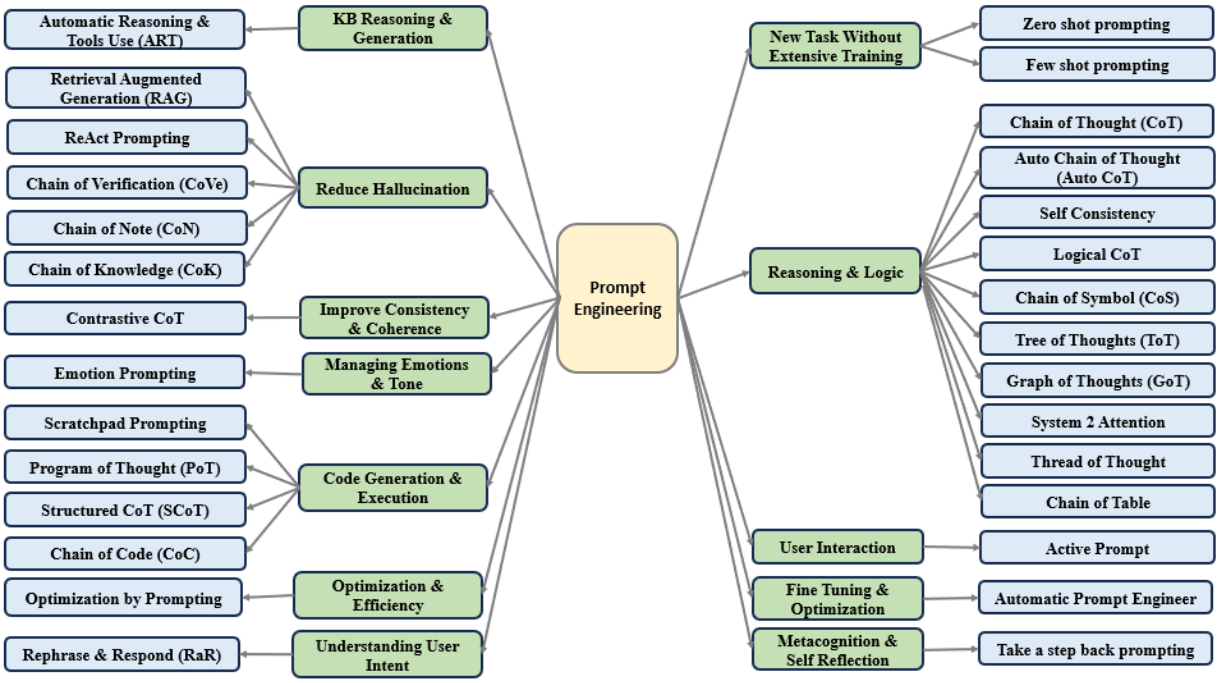

Figure 2.5: Prompt Engineering Techniques

References - (Sahoo et al., 2024)

New Tasks Without Extensive Training: Zero-shot prompting, pioneered by (Radford et al., 2019), enables large language models to tackle tasks without explicit training data by leveraging task descriptions within prompts. This technique showcases the models' ability to

generalize and adapt based on existing knowledge. Conversely, few-shot prompting, introduced by (Brown et al., 2020), improves performance on complex tasks by providing models with a limited number of input-output examples. However, it requires careful prompt engineering to prevent biases and ensure optimal performance, particularly in large pre-trained models like GPT-3.

Reasoning & Logic: A multitude of prompting techniques have emerged to augment the capabilities of Large Language Models (LLMs) across diverse domains, each addressing specific challenges and scenarios. For instance, Chain of Thought (CoT) Prompting, as proposed by (Wei et al., 2022), guides models through a sequence of prompts to foster systematic reasoning and problem-solving, building upon each prompt to lead the model toward the desired outcome. In contrast, Auto-CoT, introduced by (Zhang et al., 2022), automates the generation of reasoning chains, thereby enhancing the model's robustness and ability to perform well in few-shot learning scenarios. Furthermore, Self-Consistency, as outlined by (Wang et al., 2022), enhances reasoning performance by diversifying the generated reasoning chains and selecting the most consistent final answer. Other techniques, such as LogiCoT (Zhao et al., 2023b), CoS (Hu et al., 2023), Tree-of-Thoughts (ToT) (Yao et al., 2023a), Graph of Thoughts (GoT) (Yao et al., 2023b), System 2 Attention (S2A)(Weston and Sukhbaatar, 2023), ThoT (Zhao et al., 2023b), and Chain-of-Table (Wang et al., 2024), each introduce unique strategies to enhance reasoning abilities, handle complex spatial relationships, manage chaotic contexts, improve attention, and optimize tabular reasoning, respectively. Together, these techniques signify a broad spectrum of approaches aimed at refining the reasoning capabilities of LLMs across various tasks and environments.

Reduce Hallucination: (Lewis et al., 2020) proposed Retrieval Augmented Generation (RAG) to integrate information retrieval into Large Language Models (LLMs), enabling accurate responses to tasks needing external knowledge. ReAct, by (Yao et al., 2022), allows LLMs to concurrently generate reasoning traces and actions, improving performance across language and decision-making tasks. (Dhuliawala et al., 2023) Chain-of-Verification (CoVe) method systematically verifies model responses, enhancing logical reasoning and reducing errors. (Yu et al., 2023) Chain-of-Note (CoN) Prompting improves RALMs' handling of irrelevant knowledge, and (Li et al., 2023d)Chain-of-Knowledge (CoK) Prompting overcomes

limitations in traditional techniques by breaking down complex tasks into coordinated steps, leading to improved performance.

User Interaction: (Diao et al., 2023) introduced Active-Prompting to improve the performance of Large Language Models (LLMs) on diverse reasoning tasks. Instead of using fixed sets of human-annotated examples like existing methods, Active-Prompt employs uncertainty-based active learning to select important questions for annotation, resulting in better performance across various complex reasoning tasks in text and code domains.

Fine Tuning and Optimization: The Automatic Prompt Engineer (APE) proposed by (Zhou et al., 2023) is an innovative method for creating prompts for Learning Language Models (LLMs). It dynamically generates and selects the most impactful prompts for specific tasks, leveraging reinforcement learning. APE has shown to exceed human-authored prompts in most cases and significantly boost LLMs' reasoning abilities, allowing them to tackle a wider range of tasks with greater efficiency.

Knowledgebase reasoning & Tools Usage: (Paranjape et al., 2023) introduced Automatic Reasoning and Tool-use (ART) to enhance the capabilities of Large Language Models (LLMs) in complex tasks by enabling multi-step reasoning and integration of external expertise. ART automates reasoning steps through structured programs, seamlessly integrates external tools, and demonstrates effectiveness on challenging benchmarks, surpassing traditional prompting techniques and matching hand-crafted demonstrations in some cases.

Improve Consistency and Coherence: Contrastive Chain-of-Thought (CCoT) Prompting, introduced by (Chia et al., 2023), enhances traditional CoT prompting for Large Language Models (LLMs) by incorporating learning from mistakes through providing both valid and invalid reasoning demonstrations alongside original prompts. CCoT demonstrates notable improvements of 4-16% in strategic and mathematical reasoning evaluations compared to traditional CoT, although questions remain about its automated generation of contrasting demonstrations for diverse problems and its applicability to other NLP tasks beyond reasoning.

Managing Emotion and Tone: (Li et al., 2023b) introduced Emotion Prompt to enhance Large Language Models' (LLMs) comprehension of emotional cues by appending 11 emotional

stimulus sentences to prompts. Experimental results showed significant improvements in LLM performance across various tasks, with Emotion Prompt demonstrating an 8.00% relative improvement in instruction induction and a remarkable 115% boost in BIG-Bench tasks.

Code Generation & Execution: Scratchpad Prompting, introduced by (Nye et al., 2021), offers a novel solution to the challenges faced by Transformer-based language models during intricate algorithmic calculations by introducing a 'scratchpad' concept. This allows models to generate intermediate tokens before providing final answers, surpassing existing methods like MBPP-aug with a success rate of 46.8%. Meanwhile, PoT prompting by (Chen et al., 2022) leverages external language interpreters to enhance numerical reasoning, demonstrating a 12% improvement on mathematical word problems and financial questions. Structured Chain-of-Thought (SCoT) prompting, proposed by (Li et al., 2023c), incorporates program structures into reasoning steps to enhance source code generation, achieving up to a 13.79% improvement. Lastly, Chain-of-Code (CoC) prompting, introduced by (Li et al., 2023a), improves language models' semantic reasoning skills through codewriting, achieving 84% accuracy on BIG-Bench Hard and a 12% gain, effective with both large and small models.

Understand User Intent: (Deng et al., 2023) introduce Rephrase and Respond (RaR) Prompting, which addresses the discrepancy between human thought processes and those of Large Language Models (LLMs). RaR allows LLMs to rephrase and elaborate on questions within a single prompt, leading to better comprehension and accuracy in responses. The two-step RaR variant, incorporating rephrasing and response LLMs, significantly enhances performance across various tasks by providing clearer semantics and resolving inherent ambiguities present in human queries. These findings provide valuable insights for improving the effectiveness of LLMs in diverse applications.

Metacognition and Self Reflection: This Step-Back prompting technique, developed by (Zheng et al., 2023) is a two-step method that enables advanced language models like PaLM-2L to engage in abstraction, extracting high-level concepts and fundamental principles from specific instances. Experiments demonstrate notable improvements in reasoning skills: MMLU Physics and Chemistry tests indicate a 7% improvement, TimeQA a 27% improvement, and MuSiQue a 7% improvement.

2.3.3.1 Role of Prompts in Guiding Models for Text Summarization

Prompts play a crucial role in guiding text summarization models by defining the summary's purpose, providing context, and instructing on style and length. They emphasize key information to ensure the generated summaries meet user needs, ultimately improving the quality and relevance of the summarization process. In essence, well-designed prompts empower text summarization models to produce accurate and customized summaries according to specific criteria.

2.3.5 Challenges and Open Problems

LLMs have revolutionized text summarization by overcoming challenges faced by traditional methods. They excel in abstractive summarization, generating concise, coherent summaries in their own words. LLMs also capture nuanced meanings and relationships within text, resulting in more informative summaries. They filter out grammatical errors and irrelevant information, enhancing the quality of the generated summaries. LLMs are scalable, processing large volumes of text and summarizing lengthy documents.

Despite the remarkable progress enabled by LLMs, several gaps and challenges remain in the field of LLMs based text summarization:

- Hallucination: LLMs can invent details or stray from source material, leading to factually incorrect summaries.
- Bias: Both training data and prompt wording can introduce bias into summaries.
- Difficulty with Context: LLMs struggle to understand complex relationships, sarcasm, or nuanced language, potentially leading to oversimplification or misrepresentation.
- Out-of-Vocabulary (OOV) Issues: LLMs may substitute similar words, altering summary meaning, especially for technical documents.
- Length Control: Balancing conciseness and important details can be challenging.
- Abstractive vs. Extractive Summarization: Abstractive summarization captures main ideas but is prone to hallucinations.
- Environmental Issues: Training and running large language models require significant computing resources, impacting cost and environmental footprint.

- Limited Domain Specificity: LLMs trained on general data may struggle with specialized domains.
- Evaluation Metrics: Developing robust metrics to assess summaries' quality and factuality remains a challenge.

2.4 Datasets for Text Summarization

The Automated Text Summarization algorithms require a large training dataset with ideal summaries (human annotated) to train the model. Many open-source dataset are available for text summarizing; some of the well-known ones are described below.

CNN/DailyMail: Initially curated by (Hermann et al., 2015) and later refined by (Nallapati et al., 2016a) is an English-language dataset with over 300k unique news articles from CNN and the Daily Mail. It consists of news articles paired with human-generated summaries, encompassing diverse topics and writing styles It supports both extractive and abstractive summarization.

DUC: This dataset was created by the Document Understanding Conference, is a standardized platform for evaluating and advancing summarization systems. It pairs news articles with human-generated summaries and serves as a benchmark in text summarization research. The National Institute of Standards and Technology facilitates this work.

Gigaword: Curated by (Rush et al., 2015), Is a large-scale collection of newswire documents linked with headline summaries and serves as an invaluable resource for text summarizing research. Gigaword has approximately four million papers gathered from numerous news agencies, and covers a wide range of themes and writing styles. Its broad coverage and consistent style make it a popular dataset for benchmarking text summarization methods.

LCTCS: Curated by (Hu et al., 2015) consists of more than 2 million authentic Chinese short text and its summary from Sino-Weibo are used to construct this dataset.

WikiHow: Curated by (Koupaee and Wang, 2018), is one of the most widely used dataset for text summarization provided by NIST. It contains article and summary pairs extracted from an online knowledge base.

Xsum: Curated by (Narayan et al., 2018) available for evaluating abstractive single-document summarization systems. It consists of news articles from BBC (2010 to 2017) with a one-sentence summary and covers a wide variety of domains.

Multi News Dataset: Curated by (Fabbri et al., 2019) is a valuable resource for multi-document summarization (MDS) research. Unlike prior datasets with just a few hundred examples, this one offers a massive collection of news articles grouped by topic. Each group comes with human-written summaries, allowing researchers to train and test models that can effectively condense information from multiple sources.

New York Times: Curated by (Sandhaus., 2008) contains over 1.8 million articles published by The New York Times between 1987 to 2007. Corpus includes content like article, summaries, written by librarians, and additional publication metadata like publication date and section

PubMed: Curated by (Cohen et al., 2018) is a vast collection of abstracts and bibliographic data from scientific journals covering topics in medicine, biology, and related life sciences. It serves as a fundamental resource for biomedical research, facilitating tasks such as information retrieval, text mining, and literature review.

2.5 Evaluation Metrics for Text Summarization

One of the main challenges in text summarization research is how to evaluate the quality and usefulness of the generated summaries. Broadly. there are two approaches followed in the evaluation of text summaries – human(manual) approach and automated evaluation metrics

2.5.1 Manual Evaluation

Human evaluators are often considered superior due to their ability to assess aspects like coherence, conciseness, readability, and content. They can also compare two summaries and specify a preference. However, human evaluation has drawbacks such as time consumption, high costs, and inconsistency. For instance, the same judge might score the same summary differently at different times. These issues make a case for using automatic summarization metrics for evaluating generated text summaries

2.5.2 Automated Evaluation Metrics

Human evaluation of text summarization is expensive, time consuming and may be biased and subjective. To alleviate these concerns a number of automated evaluation metrics are developed over past two decades.

BLEU

BLEU Score (Papineni et al., 2002): An IBM-invented metric that compares the n-grams of machine-translated sentences to those of human-translated sentences. It counts the number of matches in a weighted fashion, with a higher match degree indicating a higher degree of similarity and a higher score. It doesn't consider intelligibility and grammatical correctness.

ROUGE

ROUGE Score (Lin, 2004): Measures the overlap of n-grams in the generated summary and one or several human-constructed reference summaries. ROUGE-1, ROUGE-2, and ROUGE-L are the most commonly used versions, with ROUGE-L measuring the longest common sub-sequence. It's popular due to its correlation with human judgments of summary quality.

METEOR

METEOR (Banerjee and Lavie, 2005): This metrics takes into account both the precision and recall while evaluating a match. It was designed to fix some problems found in the BLEU metric and to correlate well with human judgment at the sentence or segment level.

BERTScore

BERTScore (Zhang et al., 2019): Leverages pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It computes precision, recall, and F1 measure, and has been shown to correlate with human judgment on sentence-level and system-level evaluation.

2.6 Summary

The literature review highlights dominance of LLMs over traditional approaches based on rules, statistics, ML, and DL for NLP tasks and identifies BERT, T5, and GPT-4 as leading LLMs, renowned for their state-of-the-art performance across tasks including text summarization.

Despite their individual strengths, a gap exists in comprehensive comparative analyses tailored specifically to text summarization. This study addresses this gap by conducting an in-depth comparative analysis of these models, systematically evaluating metrics such as summarization quality, coherence, and computational efficiency. Furthermore, the research investigates the impact of factors like fine-tuning strategies and prompt engineering on their performance in text summarization tasks. Through rigorous experimentation, this study aims to enhance understanding of LLMs in text summarization and aid practitioners and researchers in selecting the most suitable model for their needs.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This section describes a systematic method for investigating and comparing the performance of various LLMs for text summarization. This entails carefully selecting relevant algorithms, LLMs, datasets, and procedures to achieve a thorough and comprehensive review process. The methodology includes a complete pipeline, from data preparation to text summary deployment and evaluation.

3.2 Algorithms & Techniques

3.2.1 Transformers

The Transformer is a deep learning model architecture developed by (Vaswani et al., 2017a)for natural language processing tasks. It uses a self-attention mechanism to efficiently capture long-range dependencies in data, allowing it to weigh the importance of different input data elements without relying on recurrent or convolutional neural networks. The Transformer architecture has now become a fundamental paradigm for a wide range of natural language processing (NLP) applications, including machine translation, text production, question answering, and others.

Technical architecture of generic transformer is depicted in the Figure 3.1. High level overview of the architecture is described below:

Self-Attention Mechanism: This is the core component of Transformer model, which allows the model to assess the relevance of distinct words in a sequence as they are processed. This approach allows the model to capture long-term dependencies efficiently.

Encoder-Decoder Structure: The Transformer model is normally made up of an encoder and a decoder. The encoder processes the input sequence, while the decoder creates the output sequence. Each encoder and decoder layer is made up of many self-attention layers, followed by position-wise feedforward neural networks.

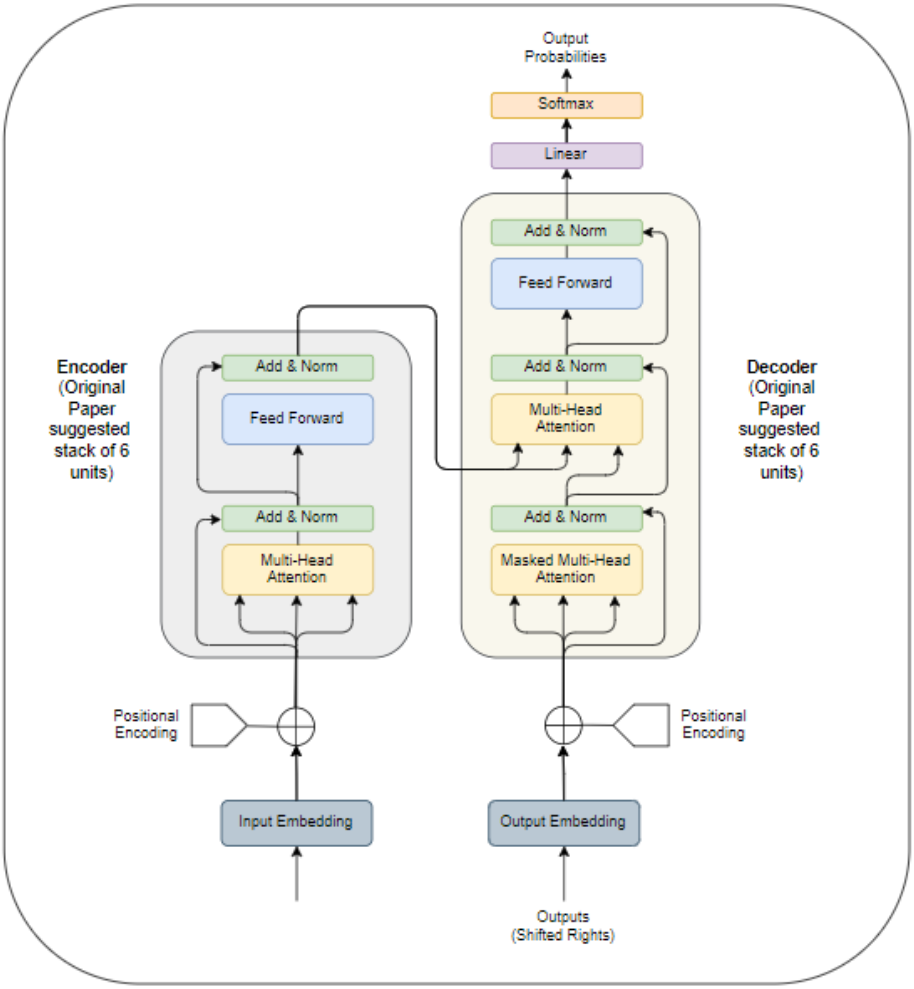

Figure 3.1: Transformer Architecture (Vaswani et al., 2017b)

Positional Encoding: This is incorporated into the Transformer model to compensate for its lack of inherent understanding of word order within a sequence. This encoding method supplements the input embeddings with positional information, enabling the model to account for the sequential arrangement of words during computation.

Multi-Head Attention: The Transformer model uses multi-head attention to improve the expressiveness of self-attention. This technique involves applying the self-attention mechanism several times in parallel, each time with a new set of learnt linear projections. This enables the model to concurrently focus on various segments of the input stream.

Feed Forward Neural Network: Following the self-attention mechanism, every position within the sequence is subjected to a position-wise feedforward neural network. This network involves two linear transformations separated by a non-linear activation function.

Layer Normalization and Residual Connections serve to stabilize the training process within deep neural networks. These techniques are applied to every sub-layer, encompassing both the self-attention and feedforward layers, across both the encoder and decoder components.

The Transformer architecture has revolutionized Natural Language Processing (NLP) by effectively capturing long-range dependencies through its self-attention mechanism. LLM variants like BERT - encoder only (Devlin et al., 2019), GPT - decoder only (Radford et al., 2019; Brown et al., 2020), and T5 - encoder – decoder (Raffel et al., 2019) have further enhanced its performance and applicability, leading to significant advancements in tasks such as machine translation, text generation, and question answering. This adaptability and versatility capability of Transformers have reshaped the NLP landscape, pushing the boundaries of NLU and NLG across diverse domains.

3.2.2 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It is a language model (LM) that can learn the representation of natural language from unlabeled text in an unsupervised way. BERT uses the transformer architecture, which consists of two main components: the encoder and the decoder. The encoder transforms the input sequence of tokens into a sequence of hidden states, and the decoder generates the output sequence from the hidden states.

As depicted in Figure 3.2, BERT architecture only uses the encoder part of the transformer and discards the decoder. The encoder is composed of a stack of N identical layers, each with two sub-layers: a multi-head self-attention layer and a feed-forward layer. The self-attention layer allows the encoder to capture the dependencies between any pair of tokens in the input, regardless of their distance. The feed-forward layer applies a non-linear transformation to each

hidden state independently. Both sub-layers are followed by a residual connection and a layer normalization.

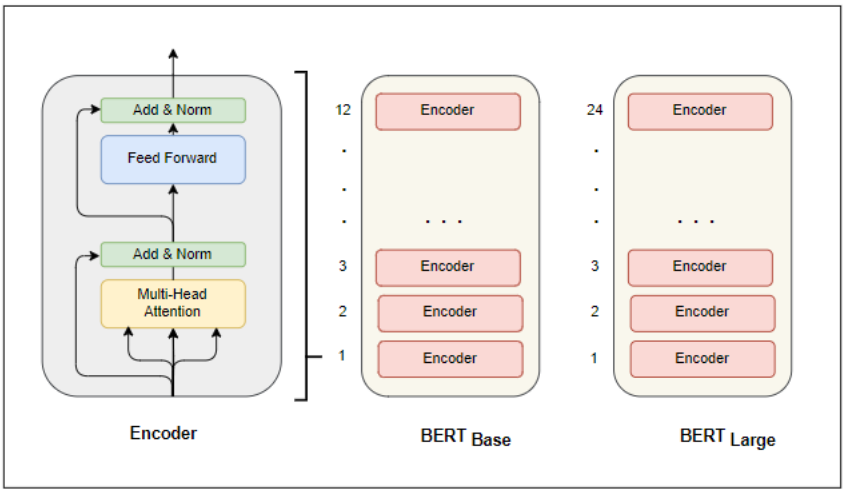

Figure 3.2: BERT Model Architecture

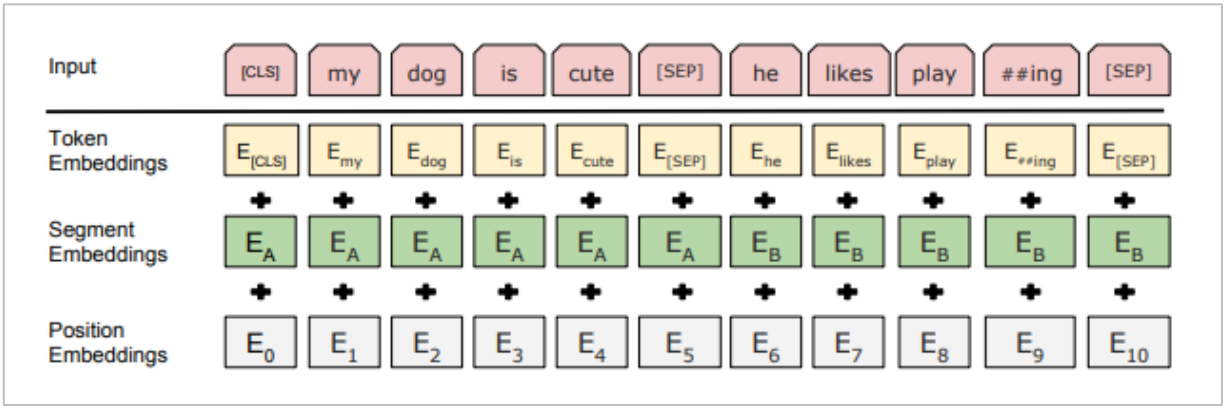

Figure 3.3: BERT Input Representation

References - (Devlin et al., 2019)

As depicted in the Figure 3.3, the input to the encoder is a sequence of tokens, each represented by an embedding vector. The token embeddings are obtained by adding three types of embeddings: word, segment, and position embeddings. Word embeddings are learned from a large vocabulary of tokens, and encode the semantic information of each token. Segment embeddings indicate whether a token belongs to the first or the second sentence in a

pair of sentences (BERT can take two sentences as input for some tasks, such as natural language inference or question answering). Position embeddings encode the order of the tokens in the sequence, and allow the model to learn the syntactic structure of the sentences.

BERT is trained on two unsupervised tasks: masked language modeling (MLM) and next sentence prediction (NSP) as depicted in Figure 3.X. MLM randomly masks some tokens in the input, and asks the model to predict the original tokens based on the context. This forces the model to learn bidirectional representations, unlike traditional LMs that only use left-to-right or right-to-left contexts. NSP takes a pair of sentences as input, and asks the model to predict whether the second sentence follows the first one in a coherent text. This helps the model to learn the relationship between sentences, and to capture the discourse-level information.

Masked LM helps BERT to understand the context within a sentence and Next Sentence Prediction helps BERT grasp the connection or relationship between pairs of sentences. Hence, training both the strategies together ensures that BERT learns a broad and comprehensive understanding of language, capturing both details within sentences and the flow between sentences.

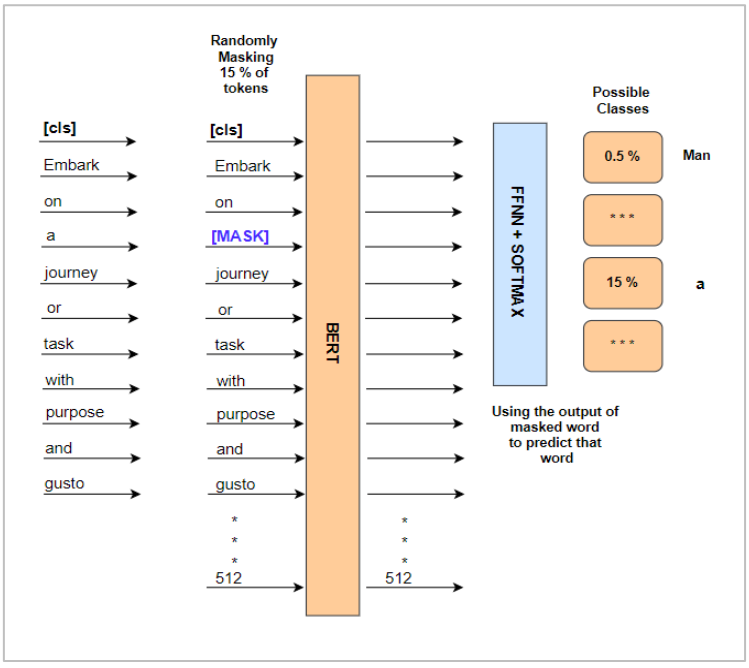

Figure 3.4: Masked Language Model

References - (Devlin et al., 2019)

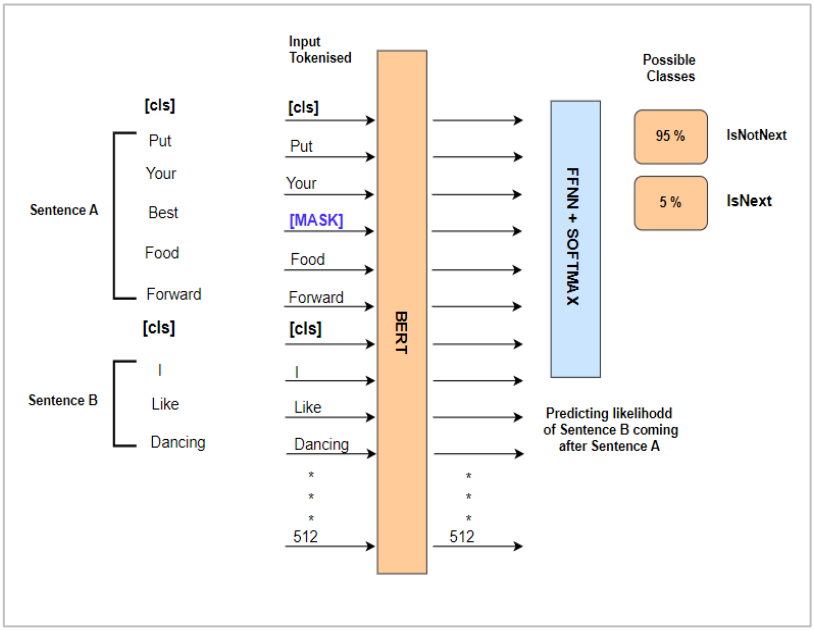

Figure 3.5: Next Sentence Prediction

References - (Devlin et al., 2019)

By pre-training BERT on a large corpus of text, such as Wikipedia (2,500M words) and Toronto BookCorpus (800M words), the model can learn general-purpose representations that capture various aspects of natural language. These representations can then be fine-tuned on specific downstream tasks, such as classification, named entity recognition, or question answering, by adding a task-specific layer on top of the pre-trained encoder and updating all the parameters with a small amount of labeled data. BERT has achieved state-of-the-art results on many natural language understanding benchmarks, such as GLUE, SQuAD, and SWAG.

Since its release, BERT has inspired a number of variations and adaptations. Here are several prominent variations of BERT and their adaptations:

- RoBERTa - Robustly Optimized BERT Approach (Liu et al., 2019)is an enhancement to the basic BERT model. It modifies key hyper parameters removes the next sentence prediction objective, and trains with more data for more iterations. These changes result in enhanced performance on a variety of NLP tasks.

- ALBERT (A Lite BERT): ALBERT strives to minimize BERT's computational requirements while retaining performance. It accomplishes this by sharing parameters between levels and utilizing cross-layer parameter sharing. This enables a large reduction in the number of parameters while retaining performance.
- DistilBERT: This is a smaller, distilled version of BERT (Sanh et al., n.d.) that retains much of its performance but has fewer parameters. It accomplishes this by utilizing a reduced model architecture and a knowledge distillation technique during training.
- BERT Large and BERT Base: BERT is available in two sizes: BERT Base (12-layer, 768-hidden-nodes, 12-attention-heads, 110M parameters) and BERT Large(24-layer, 1024-hidden-nodes, 16-attention-heads, 340M parameters). BERT Large has more layers and parameters compared to BERT Base, allowing it to capture more complex patterns but requiring more computational resources.
- ELECTRA (Efficiently Learning an Encoder that Accurately Classifies Token Replacements): Electra proposes a new pre-training task known as replaced token detection, in which a small percentage of input tokens are replaced with incorrect ones and the model is trained to distinguish between the original and replaced tokens.This approach is more computationally efficient than BERT's masked language modeling task.
- Language BERT: Several language specific models are developed over time, prominent among them are BERTje(Dutch), CamemBERT(French) and KoBERT(Korean) models

3.2.3 T5

The T5 model, developed by Google researchers (Raffel et al., 2019), is a sophisticated language model that operates on a Transformer architecture, akin to BERT, but with notable distinctions. It comprises both encoder and decoder components, allowing bidirectional processing, which is particularly advantageous for tasks necessitating input-output mapping, like language translation and text summarization.

Distinctively, T5 adopts a "text-to-text" framework, treating all NLP tasks as text mapping challenges, where inputs and outputs are represented as text strings. This unified approach enables T5 to handle a wide array of tasks by simply modifying input and output representations, rendering it remarkably versatile and adaptable.

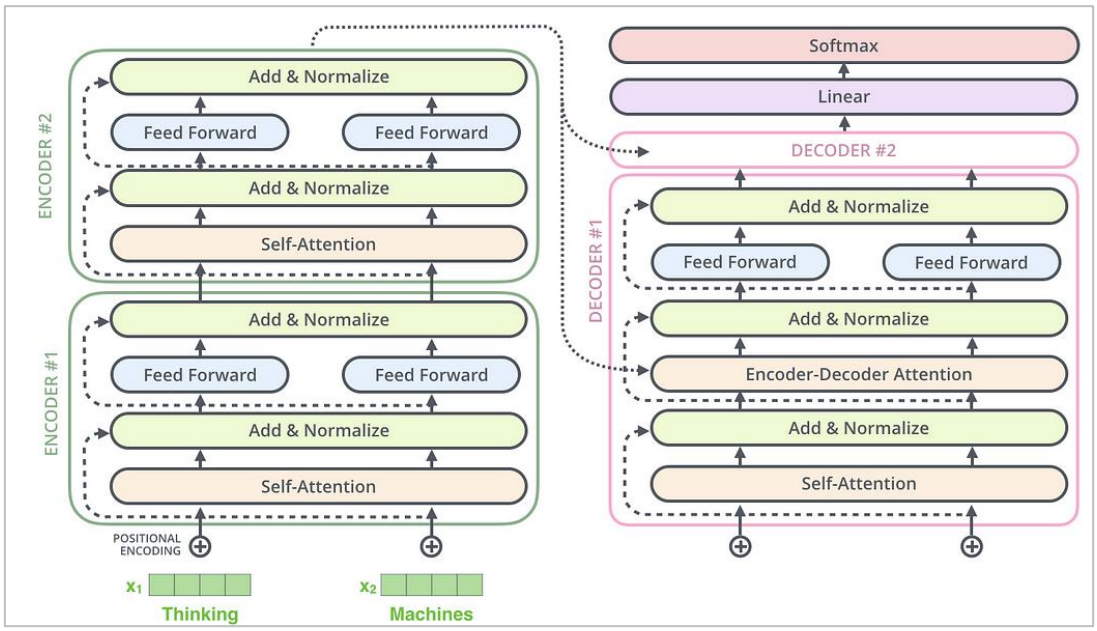

Figure 3.6: T5 Model Architecture

References - (Raffel et al., 2019)

As depicted in the Figure 3.6, the T5 model's architecture is almost identical to the original Transformer presented by Vaswani et al. Both the encoder and decoder include 12 blocks. This model contains 220 million parameters. Only a few changes have been made to the design, such as removing the Layer norm bias and moving the layer normalization outside of the residual path. T5 has a different position embedding approach. T5 was trained on 750 GB Colossal Clean Crawled Corpus(C4) dataset. T5 follows same technique of unsupervised pre-training and supervised fine-tuning. During pre-training, T5 learns to generate target text from input text across various text-to-text tasks using unsupervised learning objectives. Subsequently, fine-tuning on task-specific data customizes T5's parameters to the specific task requirements.

An innovative aspect of T5's pre-training is its exposure to a diverse array of tasks concurrently, including translation, summarization, and question answering. This multifaceted approach

equips the model with comprehensive representations that effectively generalize across tasks, facilitating transfer learning.

T5 employs a shared vocabulary for input and output tokens, ensuring seamless task mapping and simplifying the model's architecture. This shared vocabulary enhances T5's versatility, making it a potent tool for a multitude of natural language understanding and generation tasks.

T5 model is available in different sizes:

- t5-small
- t5-base
- t5-large
- t5-3b
- t5-11b

Based on the initial T5 model, Google has produced some follow-up works:

- T5v1.1: This is an enhanced version of T5 with certain architectural changes, and it is pre-trained just on C4 without any supervised jobs.
- mT5: This is multilingual T5 model, pre-trained on the mC4 corpus (101 languages)
- byT5: This is trained on byte sequences rather than Sentence Piece sub word token sequences.
- UL2: This model has been pre-trained on a variety of denoising targets.
- Flan-T5: Uses pretraining strategy that relies on prompting and these models are trained on the Flan datasets.
- FLan-UL2: This model was fine-tuned with the "Flan" prompt tuning and dataset collection.
- UMT5: This is a multilingual model trained on an upgraded and renewed mC4 multilingual corpus of 29 trillion characters from 107 languages and utilizes a new sampling strategy called UniMax.

3.2.4 GPT 4

GPT 4 represents a cutting-edge language model built upon transformer architecture. Utilizing the Mixture of Experts (MoE) architecture, GPT 4 stands out as a multimodal model capable of processing both image and text inputs and generating corresponding outputs. Its design includes an encoder, decoder, and attention mechanism, all strategically employed to focus on pertinent inputs and produce relevant outputs. The design details of GPT 4 are still not disclosed by OpenAI.

GPT 4 outperforms previous LLMs as well as the majority of cutting-edge systems as of 2023. On the MMLU benchmark, an English-language set of multiple-choice questions encompassing 57 themes, GPT 4 not only surpasses previous models in English, but also in other languages (OpenAI et al., 2023).

Table 3. 1: GPT Model Variants

GPT Model	Description	Context Window	Training Data
gpt-4-0125-preview	The most recent GPT-4 model was designed to reduce instances of "laziness" in which the model fails to complete a task. It returns a maximum of 4,096 output tokens.	128,000 tokens	Up to Dec 2023
gpt-4-turbo-preview	Currently refers to gpt-4-0125-preview	128,000 tokens	Up to Dec 2023
gpt-4-1106-preview	GPT-4 Turbo model, which includes enhanced instruction following, JSON mode, reproducible outputs, parallel function calling, and more. It returns a maximum of 4,096 output tokens.	128,000 tokens	Up to Apr 2023
gpt-4	Currently refers to gpt-4-0613.	8,192 tokens	Up to Sep 2021
gpt-4-0613	Snapshot of GPT-4 dated June 13th, 2023 with better function calling support	8,192 tokens	Up to Sept 2021
gpt-4-32k	Currently points to gpt-4-32k-0613.	32,768 tokens	Up to Sep 2021
gpt-4-32k-0613	Snapshot of gpt-4-32k dated June 13th, 2023, with better function calling support. This model was never generally used in favor of GPT-4 Turbo.	32,768 tokens	Up to Sep 2021
gpt-3.5-turbo-0125	The most recent GPT-3.5 Turbo model, with improved accuracy in replying in specified formats and bug fixes. Returns up to 4,096 output tokens.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo	Currently refers to gpt-3.5-turbo-0125	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-1106	GPT-3.5 Turbo model includes enhanced instruction following, JSON mode, reproducible outputs, parallel function calling, & more. Returns up to 4,096 output tokens.	16,385 tokens	Up to Sep 2021

gpt-3.5-turbo-instruct	Similar capabilities to GPT-3 models. Compatible with the legacy Completions endpoint.	4,096 tokens	Up to Sep 2021

Reference: https://platform.openai.com/docs/models/overview

3.3 Language Model Adaptation Approach

To adapt a LLM for text summarization, fine-tuning the pre-trained model with a target dataset is crucial, allowing it to learn to generate concise summaries. Additionally, incorporating effective prompts during inference can guide the model to focus on key information and produce coherent summaries, further enhancing its performance in summarization tasks. Through this combined approach, the adapted LLM can effectively generate high-quality summaries tailored to specific needs.

Table 3. 2: LLMs Adaptation Technique for Text Summarization

Adaptation Technique	**BERT**	**T5**	**GPT 4**
Pre-Training	Leverage pre trained BERT models like 'bert_base_uncased'	Leverage pre trained T5 models like 't5_small'	Leverage pre trained GPT model like 'gpt-4-turbo'
Fine Tuning	Fine tune pre trained model on CNN / DailyMail dataset	Fine tune pre trained model on CNN / DailyMail dataset	Leverage pre trained model with few shot training prompts to guide the model on summarization task
Prompt Engineering	Not applicable	Not applicable	Develop custom prompts for text summarization
Domain Adaptation	Fine tuning BERT model on domain specific dataset (CNN DailyMail dataset) incorporates learning domain specific features for text summarization	Fine tuning T5 model on domain specific dataset (CNN DailyMail dataset) incorporates learning domain specific features for text summarization	Leverage prompts with few shots training to guide the model for generating suitable text summaries

3.4 Methodology

Text summarization methodology tailored to the domain specific dataset, leveraging advanced language models such as BERT, T5, and GPT-4 are depicted in Figure 3.7. Initial steps involve data preprocessing to prepare the dataset for training and evaluation. Fine-tuning of

pre-trained LLMs on the dataset allows the models to learn summarization nuances. Prompt engineering techniques are employed, particularly for GPT models, to guide the summarization process effectively. Finally, evaluation metrics like ROGUE and BLEU gauge the quality and accuracy of generated summaries, illustrating the efficacy of advanced LLMs in enhancing text summarization tasks.

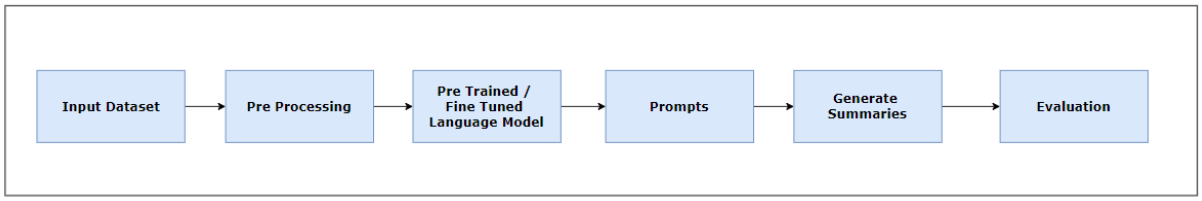

Figure 3.7: Text Summarization High Level Approach

3.4.1 Fine Tuning Pre-Trained LM

Flowchart for the fine tuning of pre-trained BERT/T5 model from HuggingFace repository using CNN / Daily Mail dataset is depicted in Figure 3.8. Description of steps followed during fine tuning a language model is explained below:

- Load pretrained LLM from repository (Hugging Face or Open AI cloud)
- Load CNN / Dailymail dataset from repository (Hugging Face)
- Pre-Process data to clean data and prepare embeddings
- Split dataset to training, validation and test dataset
- Initialize the ML model parameters
- Run the fine-tuning steps on pre trained model using training dataset
- Evaluate the model performance on validation dataset. If the performance is not adequate, fine tune the hyper parameter and rerun the fine-tuning steps till model reaches acceptable performance
- Persist the fine-tuned model for later usage
- Bench mark the fine-tuned model against the test dataset and tabulate the results for evaluation

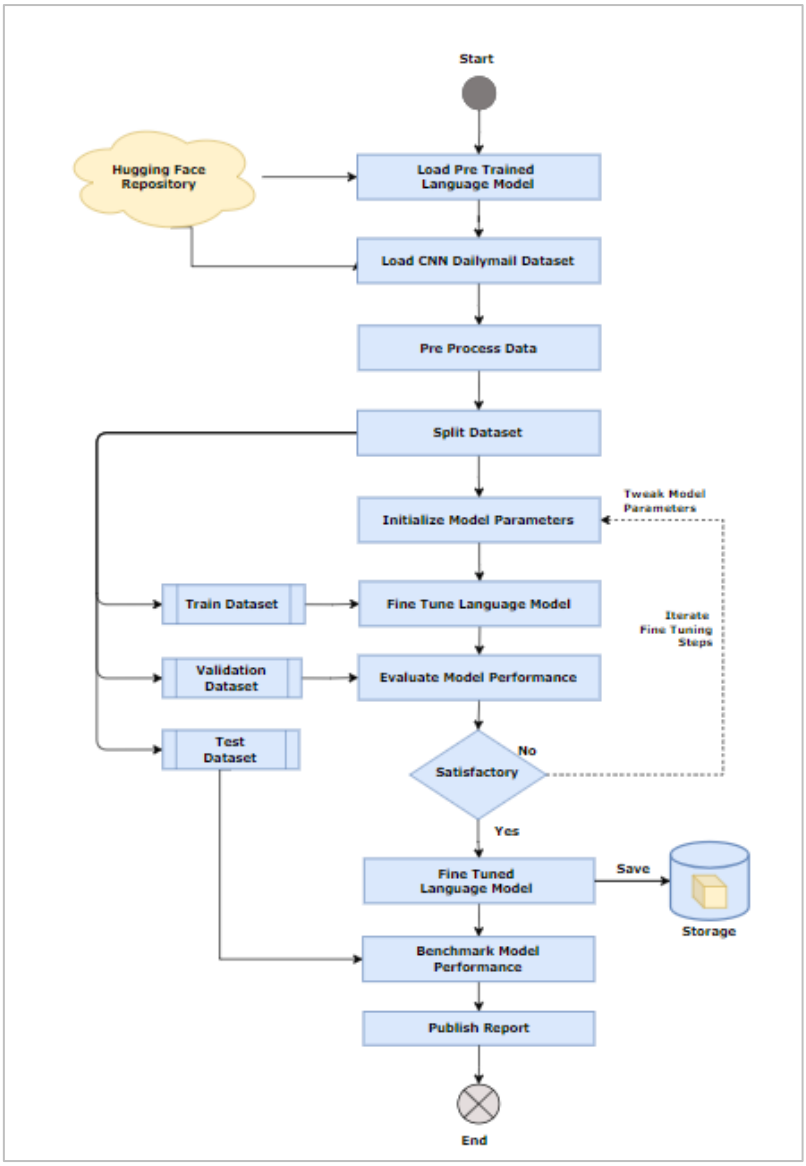

Figure 3.8: Fine Tuning a Pre-Trained Language Model

3.4.2 End-to-End Pipeline

The end-to-end pipeline for the methodology followed to evaluate and compare the efficacy of LLMs for text summarization is depicted in Figure 3.9.

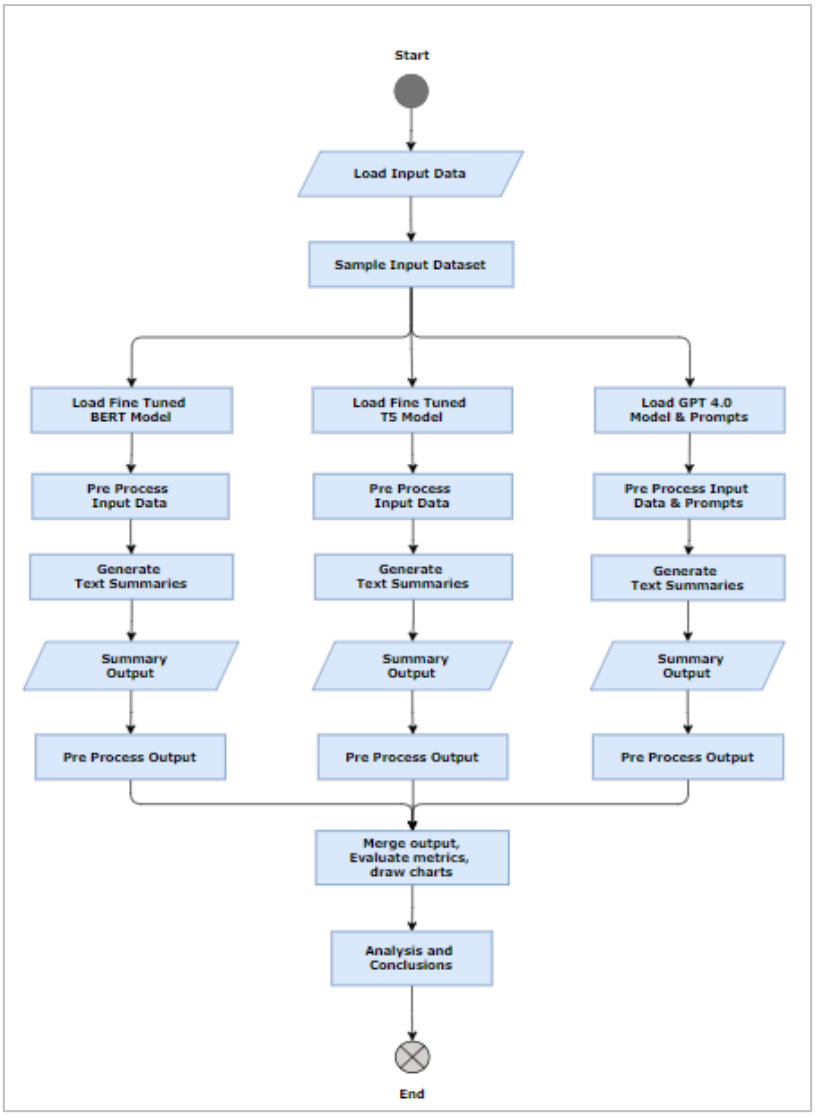

Figure 3.9: End-to-End Pipeline for Text Summarization and Evaluation

Description of the steps followed in the pipeline for analysis of LLMs are described below:

- Load the CNN/Dailymail test dataset
- Randomly Sample test dataset and extract few records (1000 records)
- Load the fine-tuned language models
- Pre-Process the sampled test dataset
- Pass the sample test data set to fine-tuned BERT, T5 and GPT models for generating summaries

- collect the generated summaries and perform any processing steps
- Save the generated summaries from each of the model
- Collect and merge the output into single data frame
- Evaluate summarization metrics like ROUGE and BLEU scores
- Analyze, compare and draw conclusions. Highlight the best performing model and its accuracy score.

3.3.2 Dataset Description

The CNN DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016a) is a widely-used benchmark in natural language processing (NLP) for tasks such as text summarization, document understanding, and content extraction. As depicted in Figure 3.10, consists of news articles and highlight sentences, with articles used as context and entities hidden in highlight sentences. The CNN news articles were created between April 2007 and April 2015, and the Daily Mail articles were written between June 2010 and April 2015. The CNN/DailyMail dataset contains over 300k unique news articles, with the current version supporting both extractive and abstractive summarization. The articles span various topics, including politics, sports, entertainment, and world events, providing diversity in content.

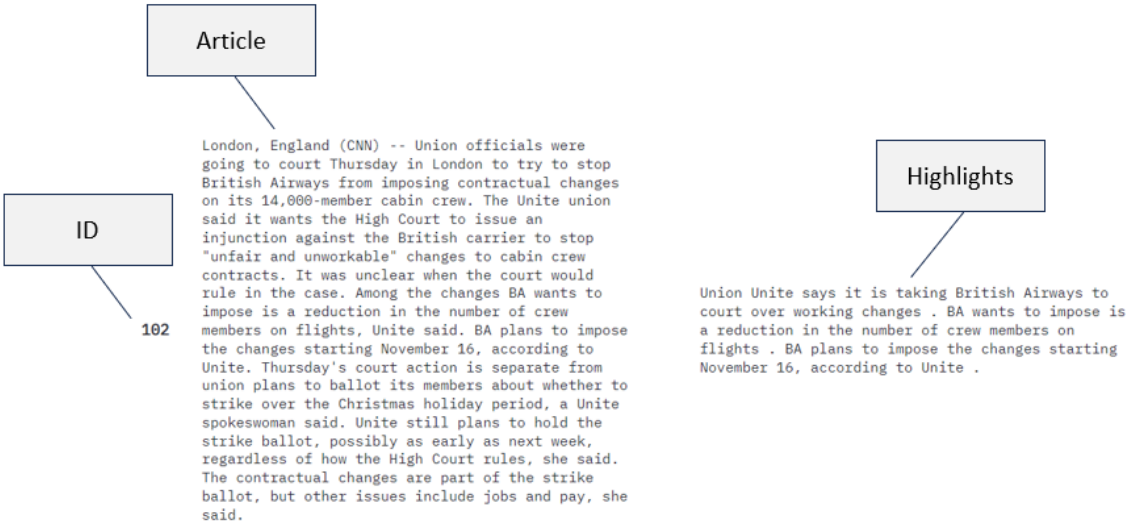

Figure 3.10: Sample Record from CNN Dailymail dataset

Data Fields:

- Id: a string containing the hash of the URL where the story was retrieved from
- Article: a string containing the body of the news article
- Highlights: a string containing the highlight of the article as written by the article author

Data Splits:

According to their scripts, the corpus consists of a total of 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs. The summaries are 53 words and 3.72 sentences long, whereas the source texts in the training set average 766 words across 29.74 sentences.

Dataset Bias:

Study conducted by (Bordia and Bowman, 2019) investigated gender bias measurement and debiasing methods using the Penn Treebank, WikiText-2, and CNN / Dailymail datasets. Based on their metric, they discover that the CNN / Dailymail dataset has a marginally smaller gender bias than the other datasets.

Since the pieces were written by and for individuals in the US and the UK, it is probable that they will highlight events that were thought to be pertinent to those populations at the time of publication and will express views that are unique to those countries.

Dataset Curators:

The data was originally collected by (Hermann et al., 2015) working for Google DeepMind. Later, (Nallapati et al., 2016a) restored the collection scripts to a summary format. They also produced both anonymized and non-anonymized versions.

Licensing Information:

The CNN / Daily Mail dataset is released under the Apache-2.0 License.

3.3.3 Data Preparation

Compared to abstractive text summarization, extractive text summarization requires additional data preprocessing steps as depicted in the Figure 3.11. Some of the steps employed to prepare the dataset for text summarization are highlighted below:

- Data Cleaning: Removal of metadata, handling of special characters, and normalization of textual content.
- Data Tokenization: This step tokenizes the articles and their corresponding summaries using the model's specific tokenizer.
- Sentence Segmentation: This step splits the articles into individual sentences.
- Padding and Truncation: This step ensures uniform length by padding or truncating tokenized sequences.
- Special Tokens: This step add model-specific special tokens like [CLS], [SEP] for BERT, or <s> and </s> for T5
- Data Formatting: This step formats the data into the model's input format, encoding tokenized sequences.
- Dataset Splitting: This step divides the data into train, validation, and test data sets.
- Save Pre-processed Data: This step saves the pre-processed data sets in an efficient format for loading during model training/evaluation.

These steps ensure the dataset is properly formatted and ready for training or fine-tuning text summarization models.

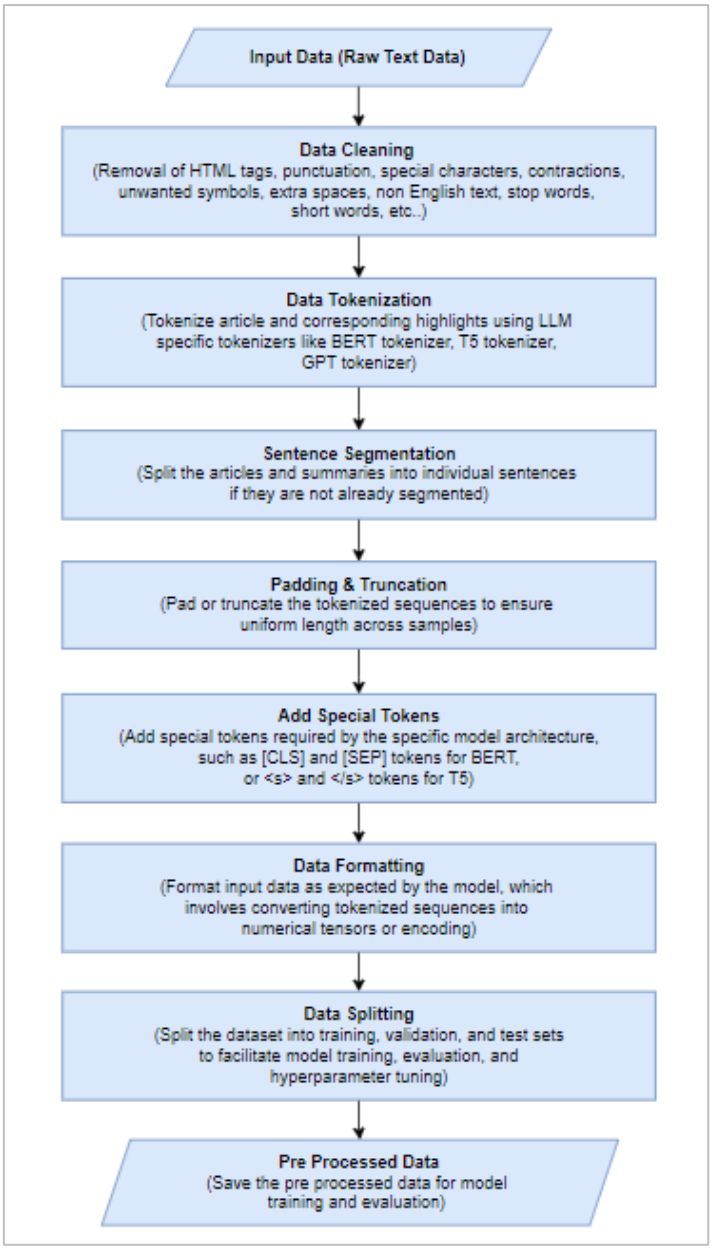

Figure 3.11: Data Preparation Steps for Text Summarization

3.3.4 Implementation Approach

The strategy outlined in Figure 3.12 illustrates various methodologies for text summarization, including Extractive, Abstractive, and Hybrid techniques. LLMs such as BERT, T5, and GPT 4 will undergo assessment within these frameworks, employing evaluation metrics such as ROUGE, BLEU, and BERTScore. Highlights of the implementation steps is as follows:

LLMs Domain Adaptation and Fine-Tuning: BERT, T5, and GPT-4 models undergo domain adaptation and fine-tuning using techniques specified in Table 3.2. This process ensures that the models are optimized for summarization tasks on the CNN Daily Mail dataset.

Baseline Model: In this study Lead3 will be used as baseline model. LLMs output will be compared against the baseline model. Performance metrics such as ROUGE, BLEU scores are used to evaluate the quality of summaries generated by each model.

Multiple Summarization Settings: The implementation approach is applied across different summarization settings, including extractive, abstractive, and hybrid approaches. Each setting explores the strengths and weaknesses of the models in capturing key information from the dataset.

Performance Comparison: The performances of BERT, T5, and GPT-4 models are compared across various summarization settings. Insights are drawn from the comparisons to understand which models perform best under different summarization paradigms.

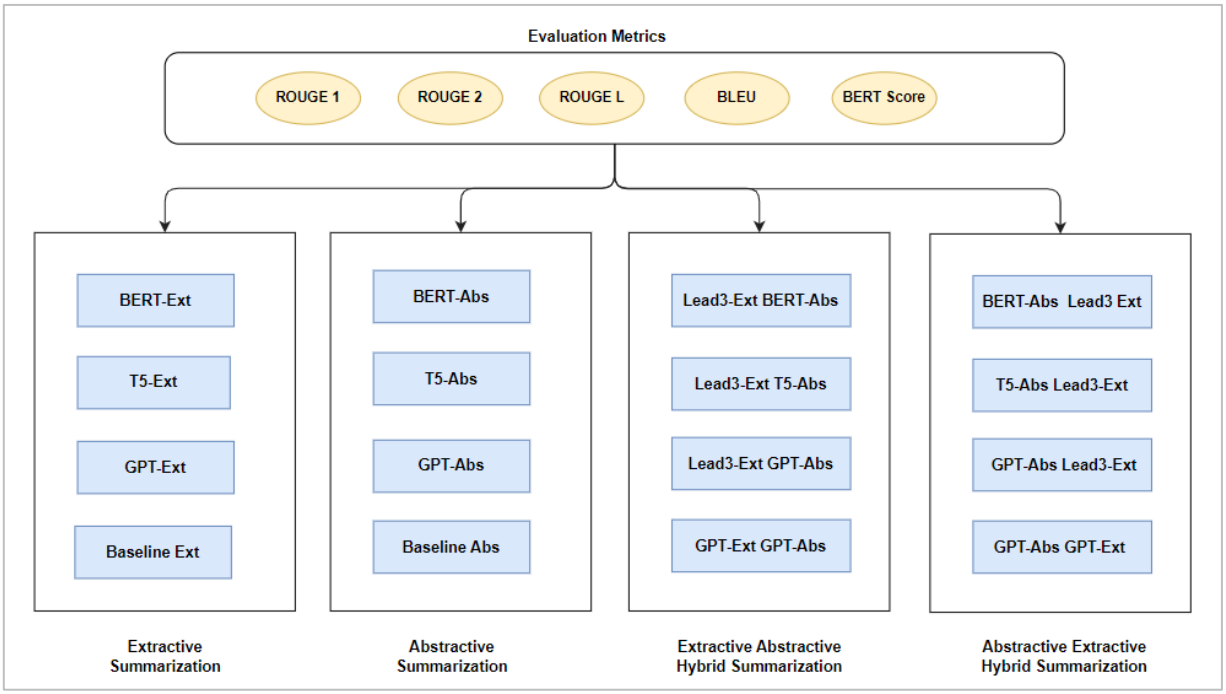

Figure 3.12: Text Summarization Implementation Approach

3.3.5 Evaluation Metrics

3.3.5.1 ROUGE

The ROUGE Score (Lin, 2004), also known as Recall-Oriented Understudy for Gisting Evaluation, comprises a collection of metrics employed to assess document translation and summarization models. It gauges the correspondence between a summary generated by a system and a set of reference summaries crafted by humans. This evaluation employs diverse techniques such as n-gram co-occurrence statistics, word overlap ratios, and other similarity metrics. Scores range from 0 to 1, with values nearing zero indicating low resemblance between the candidate output and the references, while scores closer to one signify a high degree of similarity.Higher ROUGE ratings suggest superior performance in producing a succinct summary or translation while maintaining important details from the source material.

ROUGE-N evaluates the overlap of n-grams between the system-generated summary and the reference summaries. For example, ROUGE-1 focuses on the overlap of unigrams (each word), while ROUGE-2 considers the overlap of bigrams (two consecutive words).

$$\mathrm{ROUGE}-1_{\mathrm{recall}} = \frac{\mathrm{UNIGRAM\ CANDIDATES} \cap \mathrm{UNIGRAM\ REFERENCES}}{|\mathrm{UNIGRAM\ REFERENCES}|}$$

$$\mathrm{ROUGE}-1_{\mathrm{precision}} = \frac{\mathrm{UNIGRAM\ CANDIDATES} \cap \mathrm{UNIGRAM\ REFERENCES}}{|\mathrm{UNIGRAM\ CANDIDATES}|}$$

$$\mathrm{ROUGE}-1_{\mathrm{F1}} = \frac{2 * \mathrm{recall} * \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}}$$

For ROUGE-2, the core formulas are the same as for ROUGE-1, with the only difference in using bigrams instead of unigrams.

ROUGE-L assesses the length of the Longest Common Subsequence (LCS) between the system-generated summary (candidates) and the reference summaries. It calculates a weighted harmonic mean (or f-measure), which combines the precision and recall scores. Unlike other ROUGE variants, ROUGE-L doesn't require consecutive matches but considers in-sequence matches.

$$\text{ROUGE} - \text{L}_{\text{recall}} = \frac{\text{LCS(CANDIDATES, REFERENCES)}}{\text{NUMBER OF WORDS IN REFERENCES}}$$

$$\text{ROUGE} - \text{L}_{\text{precision}} = \frac{\text{LCS(CANDIDATES, REFERENCES)}}{\text{NUMBER OF WORDS IN CANDIDATES}}$$

$$\text{ROUGE} - \text{L}_{\text{F1}} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

3.3.5.2 BLEU

The BLEU score (Papineni et al., 2002), a widely-used metric in machine translation tasks, assesses the quality of machine-generated translations by comparing them to a set of reference translations crafted by human translators. It employs n-grams, which are contiguous sequences of n words, to measure the similarity between the machine-translated text and the reference translations. Typically, unigrams (single words), bigrams (two-word sequences), trigrams (three-word sequences), and so forth are utilized as n-grams.

By examining the precision of the machine-generated translation's n-grams against the reference translations, the BLEU score determines the accuracy of the translation. To account for translations that are shorter than the reference translations, a brevity penalty is then applied to the precision.

The following is the formula for BLEU score referred from (Papineni et al., 2002)

$$BLEU = BP * \exp(\sum_{n=1}^{N} W_n * \ log\ p_n\ )$$

$$\text{BP} = \begin{cases} 1 & \text{if c} > \text{r} \\ e^{(1-\text{r/c})} & \text{if c} < \text{r} \end{cases}$$

$$Pn = \frac{\sum_{C \epsilon \{candidates\}} * \sum_{n-gram \epsilon \{C\}} Count_{clip}\ (n-gram)}{\sum_{C^1 \epsilon \{candidates\}} * \sum_{n-gram^1 \epsilon \{C^1\}} Count\ (n-gram^1)}$$

Where:

BP = Brevity Penalty

Pn = Precision of n-grams

3.3.5.3 METEOR

The METEOR score (Banerjee and Lavie, 2005) is a metric used to evaluate the quality of machine translation outputs. It stands for Metric for Evaluation of Translation with Explicit Ordering. Unlike other metrics such as BLEU and ROUGE, which primarily focus on lexical overlap, METEOR incorporates additional linguistic and semantic features to assess translation quality.

METEOR calculates a score by aligning the machine-generated translation with one or more reference translations and then measuring the similarity between them. It considers various factors such as word order, stemming, synonymy, and word-to-word matches. Additionally, METEOR incorporates stemming and synonymy matching to account for variations in word forms and synonyms.

Overall, the METEOR score provides a comprehensive evaluation of machine translation quality by considering both lexical and semantic aspects of the translation. It is widely used in research and development of machine translation systems to assess their effectiveness in producing accurate and fluent translations.

M = F_{mean} (1- p)

$$F_{mean} = \frac{10PR}{R+9P}$$

$$\text{P} = \frac{m}{w_t}, \quad \text{R} = \frac{m}{w_r}$$

$$\text{p} = 0.5 \left(\frac{c}{u_m}\right)^3$$

m = Number of unigrams in summaries

P = Unigram precision

R = Unigram recall

w_r = Number of unigrams in reference

w_t = Number of unigrams in summaries

p = penalty

u_m = Number of mapped unigrams

c = Number of chunks

3.3.5.4 BERT SCORE

BERTScore (Zhang et al., 2019) is a contextual embedding-based metric used to assess the output quality of machine-generated text in activities like machine translation and text summarization. In contrast to conventional metrics such as BLEU and ROUGE, BERTScore measures the semantic similarity between the generated and reference text by utilizing contextual embeddings from BERT, hence offering a more precise evaluation of text quality.

The equations for calculating Precision, Recall, and F1 Score are shown in below equations. Each token x is matched with most similar token $\hat{\text{X}}$ and vice-versa for calculating Recall and Precision respectively.

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

3.4 Tools

3.4.1 Software

The software used for this project are listed below:

- Coding platform: Python 3.9
- Key Python Packages used for this project:
    - Anaconda/Jupyter-notebook (v6.5.4): Web-based interactive development environment for AI ML applications.
    - Hugging face Library (v0.15.2): Library for pretrained language models.
    - Pandas (v2.0.3): Python package for manipulation of tabular data
    - Numpy (v1,24.3): Python package for working with multi dimension arrays and matrices.
    - PyTorch(v2.1.2)
    - Tensorflow(2.15.0)
    - Openai (v1.14.3): A python package provided by OpenAI to access the OpenAI model APIs.
    - Evaluate (v0.4.1): A python library for evaluating ML models.
- Microsoft Word for thesis documentation
- Microsoft Excel for documentation of research plan
- Draw.io for drawing figures used in this study
- Chrome web browser for accessing Jupyter note books

3.4.2 Hardware

The hardware used for this project are listed below:

- Laptop/Workstation with following configuration
    - RAM: 32 GB
    - GPU: Nvidia RTX3060 - 6 GB
- Google Collab Pro infrastructure with following configuration
    - RAM: 32 GB

- GPU: Nvidia RTX3060 - 6 GB

3.5 Summary

This chapter covered a comprehensive overview of the algorithms and techniques employed in the study, focusing on the utilization of advanced LLMs for text summarization tasks. It also described end-to-end implementation pipeline, covering data collection, preprocessing methodologies leveraging libraries such as Hugging Face, OpenAI and adaptation techniques such as fine-tuning pre-trained LLMs on GPU-accelerated systems. Moreover, it delves into prompt engineering strategies aimed at guiding accurate summaries and optimizing model performance. Evaluation metrics including ROUGE, BLEU are discussed for assessing the quality of generated summaries. Additionally, the chapter outlines the hardware infrastructure utilizing GPUs and software tools such as PyTorch and Hugging Face libraries. Chapter 4 further details the practical implementation of the study, providing insights into the execution and results of the research methodology outlined in this chapter.

REFERENCES

Abhishek Kumar Singh, Gupta Manish and Varma Vasudeva, (2019) Unity in Diversity: Learning Distributed Heterogeneous Sentence Representation for Extractive Summarization. [online] Available at: http://arxiv.org/abs/1912.11688.

Alami, N., Mallahi, M. El, Amakdouf, H. and Qjidaa, H., (2021) Hybrid method for text summarization based on statistical and semantic treatment. *Multimedia Tools and Applications*, 8013, pp.19567–19600.

Alcantara, T.H.M., Krütli, D., Ravada, R. and Hanne, T., (2023) Multilingual Text Summarization for German Texts Using Transformer Models. *Information (Switzerland)*, 146.

Al-Radaideh, Q.A. and Bataineh, D.Q., (2018) A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. *Cognitive Computation*, 104, pp.651–669.

Amplayo, R.K. and Lapata, M., (2021) *Informative and Controllable Opinion Summarization*.

Anand, D. and Wagh, R., (2022) Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University - Computer and Information Sciences*, 345, pp.2141–2150.

Annapurna P Patil, Shivam Dalmia, Syed Abu Ayub Ansari and Tanay Aul, (2014) *Automatic Text Summarizer*.

Ansamma John and M Wilscy, (n.d.) *RANDOM FOREST CLASSIFIER BASED MULTI-DOCUMENT SUMMARIZATION SYSTEM*.

Azhari, M. and Kumar, Y.J., (2017) Improving text summarization using neuro-fuzzy approach. *Journal of Information and Telecommunication*, 14, pp.367–379.

Ballout, M., Krumnack, U., Heidemann, G. and Kühnberger, K.-U., (2023) Investigating Pre-trained Language Models on Cross-Domain Datasets, a Step Closer to General AI. [online] Available at: http://arxiv.org/abs/2306.12205.

Bandari, S. and Bulusu, V.V., (2023) Hybrid Optimization Based Hindi Document Summarization Using Deep Learning Technique. *International Journal on Recent and Innovation Trends in Computing and Communication*, 116, pp.94–102.

Banerjee, S. and Lavie, A., (2005) *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. [online] Available at: https://aclanthology.org/W05-0909.pdf [Accessed 3 Feb. 2024].

Banko, M., Mittal, V.O. and Witbrock, M.J., (n.d.) *Headline Generation Based on Statistical Translation*.

Barzilay, R. and Lapata, M., (2008) *Modeling Local Coherence: An Entity-Based Approach*.

Barzilay, R. and Lee, L., (2003) *Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment*. [online] Available at: https://arxiv.org/abs/cs/0304006 [Accessed 2 Feb. 2024].

Barzilay, R. and Mckeown, K.R., (2005) *Sentence Fusion for Multidocument News Summarization*.

Basyal, L. and Sanghvi, M., (2023) Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models. [online] Available at: http://arxiv.org/abs/2310.10449.

Bhat, I.K., Mohd, M. and Hashmy, R., (2018) SumItUp: A Hybrid Single-Document Text Summarizer. In: *Advances in Intelligent Systems and Computing*. Springer Verlag, pp.619–634.

Binwahlan, M.S., Salim, N. and Suanmali, L., (2010) Fuzzy swarm diversity hybrid model for text summarization. *Information Processing and Management*, 465, pp.571–588.

Bishop, J.A., Xie, Q. and Ananiadou, S., (2022) *GenCompareSum: a hybrid unsupervised summarization method using salience*. [online] Available at: https://commoncrawl.org.

Bordia, S. and Bowman, S.R., (2019) *Identifying and Reducing Gender Bias in Word-Level Language Models*. [online] Available at: https://github.com/BordiaS/language-model-bias.

Brandow, R., Mitze, K. and Rau, L.E., (1995) *AUTOMATIC CONDENSATION OF ELECTRONIC PUBLICATIONS BY SENTENCE SELECTION*. *Information Processing & Management*, .

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020) Language Models are Few-Shot Learners. [online] Available at: http://arxiv.org/abs/2005.14165.

Cao, Z., Li, W., Wei, F. and Li, S., (2018) *Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization*. [online] Available at: https://aclanthology.org/P18-1015.pdf [Accessed 2 Feb. 2024].

Carbonell, J. and Goldstein, J., (1998) *The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries*.

Chen, B., Zhang, Z., Langrené, N. and Zhu, S., (2023) Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review.

Chen, W., Ma, X., Wang, X. and Cohen, W.W., (2022) Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks.

Chen, W., Ramos, K., Mullaguri, K.N. and Wu, A.S., (2021) Genetic Algorithms For Extractive Summarization. [online] Available at: http://arxiv.org/abs/2105.02365.

Chen, Y.-C. and Bansal, M., (2018) Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. [online] Available at: http://arxiv.org/abs/1805.11080.

Cheng, J. and Lapata, M., (2016) Neural Summarization by Extracting Sentences and Words. [online] Available at: http://arxiv.org/abs/1603.07252.

Chhabra, A., Askari, H. and Mohapatra, P., (2024) Revisiting Zero-Shot Abstractive Summarization in the Era of Large Language Models from the Perspective of Position Bias. [online] Available at: http://arxiv.org/abs/2401.01989.

Chia, Y.K., Chen, G., Tuan, L.A., Poria, S. and Bing, L., (2023) Contrastive Chain-of-Thought Prompting.

Chintan Shah and Dr. Anjali Jivani, (2018) *A Hybrid Approach of Text Summarization Using Latent Semantic Analysis and Deep Learning*.

Cho, S., Lebanoff, L., Foroosh, H. and Liu, F., (n.d.) *Improving the Similarity Measure of Determinantal Point Processes for Extractive Multi-Document Summarization*. [online] Association for Computational Linguistics. Available at: https://github.

Chopra, S., Auli, M. and Rush, A.M., (2016) *Abstractive Sentence Summarization with Attentive Recurrent Neural Networks*. [online] Available at: https://arxiv.org/pdf/1509.00685.pdf [Accessed 2 Feb. 2024].

Chorowski, J. and Bahdanau, D., (2015) *Attention-Based Models for Speech Recognition*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf [Accessed 2 Feb. 2024].

Cohn, T. and Lapata, M., (2009) *Sentence Compression as Tree Transduction*. *Journal of Artificial Intelligence Research*, .

Daumé, H., Knight, K., Langkilde-Geary, I., Marcu, D. and Yamada, K., (2002) *The Importance of Lexicalized Syntax Models for Natural Language Generation Tasks*. [online] Available at: https://aclanthology.org/W02-2102.pdf [Accessed 2 Feb. 2024].

Deng, Y., Zhang, W., Chen, Z. and Gu, Q., (2023) Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves.

Devlin, J., Chang, M.-W., Lee, K., Google, K.T. and Language, A.I., (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] Available at: https://aclanthology.org/N19-1423.pdf [Accessed 2 Feb. 2024].

Dey, R. and Salem, F.M., (2017) Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A. and Weston, J., (2023) Chain-of-Verification Reduces Hallucination in Large Language Models.

Diao, S., Wang, P., Lin, Y. and Zhang, T., (2023) Active Prompting with Chain-of-Thought for Large Language Models.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W., (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. [online] Available at: https://arxiv.org/pdf/2005.14165.pdf [Accessed 2 Feb. 2024].

Du, Y. and Huo, H., (2020) News Text Summarization Based on Multi-Feature and Fuzzy Logic. *IEEE Access*, 8, pp.140261–140272.

Edmundson, H.P., (1969) *New Methods in Automatic Extracting*. [online] Available at: https://dl.acm.org/doi/abs/10.1145/321510.321519 [Accessed 2 Feb. 2024].

El-Kassas, W.S., Salama, C.R., Rafea, A.A. and Mohamed, H.K., (2021) *Automatic text summarization: A comprehensive survey*. *Expert Systems with Applications*, .

Fabbri, A.R., Li, I., She, T., Li, S. and Radev, D.R., (2019) Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. [online] Available at: http://arxiv.org/abs/1906.01749.

Falke, T., Ribeiro, L.F.R., Ajie Utama, P., Dagan, I. and Gurevych, I., (2019) *Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference*. [online] Association for Computational Linguistics. Available at: https://aclanthology.org/P18-1015.pdf [Accessed 2 Feb. 2024].

Fattah, M.A., (2014a) A hybrid machine learning model for multi-document summarization. *Applied Intelligence*, 404, pp.592–600.

Fattah, M.A., (2014b) A hybrid machine learning model for multi-document summarization. *Applied Intelligence*, 404, pp.592–600.

Fattah, M.A. and Ren, F., (2009) GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech and Language*, 231, pp.126–144.

Ferreira, R., de Souza Cabral, L., Freitas, F., Lins, R.D., de França Silva, G., Simske, S.J. and Favaro, L., (2014) A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, 4113, pp.5780–5787.

Foroutan, N., Romanou, A., Massonnet, S., Lebret, R. and Abererécole, K., (2022) *Multilingual Text Summarization on Financial Documents*. [online] Available at: https://github.com/cambridgeltl/mirror-bert.

Galgani, F., Compton, P. and Hoffmann, A., (2012) *Combining Different Summarization Techniques for Legal Text*. [online] Available at: http://www.austlii.edu.au/.

Gambhir, M. and Gupta, V., (2017) Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 471, pp.1–66.

Ganesan, K., Zhai, C. and Han, J., (2010) *Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions*. [online] Available at: http://timan.cs.uiuc.edu/.

Garner, R., (1982) Efficient Text Summarization Costs and Benefits. *The Journal of Educational Research*, 755, pp.275–279.

Gatt, A. and Reiter, E., (2009) *SimpleNLG: A realisation engine for practical applications*. [online] Available at: http://lexsrv3.nlm.nih.gov/.

Gehrmann, S., Deng, Y. and Rush, A.M., (2018) Bottom-Up Abstractive Summarization. [online] Available at: http://arxiv.org/abs/1808.10792.

Gemini Team ., (2023) Gemini: A Family of Highly Capable Multimodal Models. [online] Available at: http://arxiv.org/abs/2312.11805.

Genest, P.-E. and Lapalme, G., (2011) *Framework for Abstractive Summarization using Text-to-Text Generation*.

Genest, P.-E. and Lapalme, G., (2012a) *Fully Abstractive Approach to Guided Summarization*. [online] Association for Computational Linguistics. Available at: www.nist.gov/tac.

Genest, P.-E. and Lapalme, G., (2012b) *Fully Abstractive Approach to Guided Summarization*. [online] Association for Computational Linguistics. Available at: www.nist.gov/tac.

Ghosh, S., Dutta, M. and Das, T., (2022) Indian Legal Text Summarization: A Text Normalisation-based Approach. [online] Available at: http://arxiv.org/abs/2206.06238.

Goularte, F.B., Nassar, S.M., Fileto, R. and Saggion, H., (2019) A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications*, 115, pp.264–275.

Gū, J., Shavarani, H.S. and Sarkar, A., (2019) Pointer-based Fusion of Bilingual Lexicons into Neural Machine Translation. [online] Available at: http://arxiv.org/abs/1909.07907.

Gunes Erkan, D.R., (2011) LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. [online] Available at: https://arxiv.org/pdf/1909.07907.pdf [Accessed 2 Feb. 2024].

Gupta, S. and Gupta, S.K., (2018) *International Journal of Computer Science and Mobile Computing A Hybrid Approach to Single Document Extractive Summarization*. [online] *International Journal of Computer Science and Mobile Computing*, Available at: www.ijcsmc.com.

Gupta, V. and Kaur, N., (2016) A Novel Hybrid Text Summarization System for Punjabi Text. *Cognitive Computation*, 82, pp.261–277.

Güran, A., Uysal, M., Ekinci, Y. and Güran, C.B., (2017) An additive FAHP based sentence score function for text summarization. *Information Technology and Control*, 461, pp.53–69.

Harabagiu Sanda and Finley Lacatusu, (2001) *Generating Single and Multi-Document Summaries with GISTEXTER*. [online] Available at: http://www-nlpir.nist.gov/projects/duc/.

Hennig Leonhard, (2009) *Topic-based Multi-Document Summarization with Probabilistic Latent Semantic Analysis*. [online] Available at: http://nltk.org.

Hermann, K.M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P., (2015) Teaching Machines to Read and Comprehend. [online] Available at: http://arxiv.org/abs/1506.03340.

Hochreiter S and Jurgen S, (1997) *Long Short-Term Memory*.

Hovy, E. and Lin, C.-Y., (1996) Automated text summarization and the SUMMARIST system. In: *Proceedings of a workshop on held at Baltimore, Maryland October 13-15, 1998* -. Morristown, NJ, USA: Association for Computational Linguistics, p.197.

Hovy, E. and Lin, C.-Y., (1998) *AUTOMATED TEXT SUMMARIZATION AND THE SUMMARIST SYSTEM*.

Hu, B., Chen, Q. and Zhu, F., (2015) *LCSTS: A Large Scale Chinese Short Text Summarization Dataset*. [online] Association for Computational Linguistics. Available at: http://www.nist.gov/tac/2015/KBP/.

Hu, H., Lu, H., Zhang, H., Song, Y.-Z., Lam, W. and Zhang, Y., (2023) Chain-of-Symbol Prompting Elicits Planning in Large Langauge Models.

Ibrahim et al., (2012) *Semantic Graph Reduction Approach for Abstractive Text Summarization*. IEEE.

Ishikawa, K., Ando, S. and Okumura, A., (n.d.) *Hybrid Text Summarization Method based on the TF Method and the LEAD Method*.

Ježek, K. and Steinberger, J., (2004) *Automatic Text Summarization (The state of the art 2007 and new challenges)*. [online] Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=818dfb5d509c0571152a175f72825dc6f94569ab [Accessed 2 Feb. 2024].

Jing, H. and Mckeown, K.R., (1999) *The Decomposition of Human-Written Summary Sentences*. [online] Available at: https://dl.acm.org/doi/pdf/10.1145/312624.312666 [Accessed 2 Feb. 2024].

Khan, A., Salim, N. and Jaya Kumar, Y., (2015) A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing Journal*, 30, pp.737–747.

Khan Atif and Naomie Salim, (2015) *Genetic Semantic Graph Approach for Multi- document Abstractive Summarization*.

Khan, R., Qian, Y. and Naeem, S., (2019) Extractive based Text Summarization Using KMeans and TF-IDF. *International Journal of Information Engineering and Electronic Business*, [online] 113, pp.33–44. Available at: http://www.mecs-press.org/ijieeb/ijieeb-v11-n3/v11n3-5.html.

Knight, K. and Marcu, D., (2000) *Statistics-Based Summarization-Step One: Sentence Compression*. [online] Available at: www.aaai.org.

Koupaee, M. and Wang, W.Y., (2018) WikiHow: A Large Scale Text Summarization Dataset. [online] Available at: http://arxiv.org/abs/1810.09305.

Kumar Meena, Y. and Gopalani, D., (2015) Evolutionary algorithms for extractive automatic text summarization. In: *Procedia Computer Science*. Elsevier B.V., pp.244–249.

Kupiec, J., Pedersen, J. and Chen, F., (1995) *A Trainable Document Summarizer*. [online] Available at: https://dl.acm.org/doi/pdf/10.1145/215206.215333 [Accessed 2 Feb. 2024].

Kurisinkel, L.J. and Chen, N.F., (2023) LLM Based Multi-Document Summarization Exploiting Main-Event Biased Monotone Submodular Content Extraction. [online] Available at: http://arxiv.org/abs/2310.03414.

Kutlu, M., Ciğir, C. and Cicekli, I., (2010) Generic text summarization for Turkish. In: *Computer Journal*. pp.1315–1323.

Lakshmi, A. and Latha, D., (2022) Automatic Text Summarization for Telugu Language. In: *4th International Conference on Recent Trends in Computer Science and Technology, ICRTCST 2021 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp.223–227.

Le, H.T. and Le, T.M., (2013) An approach to abstractive text summarization. In: *2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*. Institute of Electrical and Electronics Engineers Inc., pp.371–376.

Lee, C.S., Jian, Z.W. and Huang, L.K., (2005) A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 355, pp.859–880.

Leskovec, J., Grobelnik, M. and Milic-Frayling, N., (2004) *Learning Sub-structures of Document Semantic Graphs for Document Summarization*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. and Ai, F., (2019) *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. [online] Available at: https://arxiv.org/pdf/2211.05100.pdf [Accessed 2 Feb. 2024].

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. and Kiela, D., (2020) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

Li, C., Liang, J., Zeng, A., Chen, X., Hausman, K., Sadigh, D., Levine, S., Fei-Fei, L., Xia, F. and Ichter, B., (2023a) Chain of Code: Reasoning with a Language Model-Augmented Code Emulator.

Li, C., Wang, J., Zhang, Y., Zhu, K., Hou, W., Lian, J., Luo, F., Yang, Q. and Xie, X., (2023b) Large Language Models Understand and Can be Enhanced by Emotional Stimuli.

Li, J., Li, G., Li, Y. and Jin, Z., (2023c) Structured Chain-of-Thought Prompting for Code Generation.

Li, W., Xiao, X., Liu, J., Wu, H., Wang, H. and Du, J., (2020) Leveraging Graph to Improve Abstractive Multi-Document Summarization. [online] Available at: http://arxiv.org/abs/2005.10043.

Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M. and Li, J., (2019) *Entity-Relation Extraction as Multi-turn Question Answering*. [online] Association for Computational Linguistics. Available at: https://github.com/tticoin/LSTM-ER/.

Li, X., Zhao, R., Chia, Y.K., Ding, B., Joty, S., Poria, S. and Bing, L., (2023d) Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources.

Lin, C.-Y., (2004) *ROUGE: A Package for Automatic Evaluation of Summaries*. [online] Available at: https://aclanthology.org/W04-1013.pdf [Accessed 2 Feb. 2024].

Lin, H. and Bilmes, J., (2010) *Human Multi-document Summarization via Budgeted Maximization of Submodular Functions*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. and Neubig, G., (2021) Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.

Liu Peter, Saleh Mohammad, Pot Etienne, Goodrich Ben, Sepassi Ryan, Kaiser Łukasz and Shazeer Noam, (2018) *GENERATING WIKIPEDIA BY SUMMARIZING LONG SEQUENCES*. [online] Available at: https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style.

Liu, Y. and Lapata, M., (2019a) Text Summarization with Pretrained Encoders. [online] Available at: http://arxiv.org/abs/1908.08345.

Liu, Y. and Lapata, M., (2019b) Text Summarization with Pretrained Encoders. [online] Available at: http://arxiv.org/abs/1908.08345.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. [online] Available at: http://arxiv.org/abs/1907.11692.

Lloret, E., Boldrini, E., Vodolazova, T., Martínez-Barco, P., Muñoz, R. and Palomar, M., (2015) A novel concept-level approach for ultra-concise opinion summarization. *Expert Systems with Applications*, 4220, pp.7148–7156.

Lloret, E., Romá-Ferri, M.T. and Palomar, M., (2013) COMPENDIUM: A text summarization system for generating abstracts of research papers. In: *Data and Knowledge Engineering*. Elsevier B.V., pp.164–175.

Luhn H P, (1958) *The Automatic Creation of Literature Abstracts*. [online] Available at: https://ieeexplore.ieee.org/abstract/document/5392672 [Accessed 2 Feb. 2024].

Lukas, (2021) Generating abstractive summaries of Lithuanian news articles using a transformer model. [online] Available at: http://arxiv.org/abs/2105.03279.

Malallah, S., Hussein Ali, Z. and Suhad Malallah, A., (n.d.) *Multi-Document Text Summarization using Fuzzy Logic and Association… Multi-Document Text Summarization using Fuzzy Logic and Association Rule Mining*.

Mandal, S., Achary, P., Phalke, S., Poorvaja, K.V.K. and Kulkarni, M., (2021) Extractive Text Summarization Using Supervised Learning and Natural Language Processing. In: *2021 International Conference on Intelligent Technologies, CONIT 2021*. Institute of Electrical and Electronics Engineers Inc.

Mani, Inderjeet. and Maybury, M.T., (1999) *Advances in automatic text summarization*. MIT Press.

Marcu, D., (1997) *The Rhetorical Parsing of Natural Language Texts*. [online] Available at: https://aclanthology.org/P97-1013.pdf [Accessed 2 Feb. 2024].

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C. and León, E., (2014) Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications*, 419, pp.4158–4169.

Mihalcea, R. and Tarau, P., (2004) *TextRank: Bringing Order into Texts*. [online] Available at: https://aclanthology.org/W04-3252.pdf [Accessed 2 Feb. 2024].

Mugi Karanja, J. and Matheka, A., (2022) A Hybrid Model for Text Summarization Using Natural Language Processing. *Open Journal for Information Technology*, 52, pp.65–80.

Munot, N. and S. Govilkar, S., (2015) Conceptual Framework for Abstractive Text Summarization. *International Journal on Natural Language Computing*, 41, pp.39–50.

Nallapati, R., Zhai, F. and Zhou, B., (2016a) SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. [online] Available at: http://arxiv.org/abs/1611.04230.

Nallapati, R., Zhou, B., santos, C.N. dos, Gulcehre, C. and Xiang, B., (2016b) Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. [online] Available at: http://arxiv.org/abs/1602.06023.

Narayan, S., Cohen, S.B. and Lapata, M., (2018) Don’t Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. [online] Available at: http://arxiv.org/abs/1808.08745.

Nye, M., Andreassen, A.J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., Sutton, C. and Odena, A., (2021) Show Your Work: Scratchpads for Intermediate Computation with Language Models.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H.W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S.P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S.S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, Ł., Kamali, A., Kanitscheider, I., Keskar, N.S., Khan, T., Kilpatrick, L., Kim, J.W., Kim, C., Kim, Y., Kirchner, J.H., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, Ł., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C.M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S.M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Peres, F. de A.B., Petrov, M., Pinto, H.P. de O., Michael, Pokorny, Pokrass, M., Pong, V.H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F.P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M.B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J.F.C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J.J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W. and Zoph, B., (2023) GPT-4 Technical Report.

O'Shea, K. and Nash, R., (2015) An Introduction to Convolutional Neural Networks.

Padmapriya, G. and Duraiswamy, K., (2020) Multi document based text summarisation through deep learning algorithm. *International Journal of Business Intelligence and Data Mining*, 164, p.459.

Pandya, V., (n.d.) *AUTOMATIC TEXT SUMMARIZATION OF LEGAL CASES: A HYBRID APPROACH*.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J., (2002) *BLEU: a Method for Automatic Evaluation of Machine Translation*. [online] Available at: https://aclanthology.org/P02-1040.pdf [Accessed 2 Feb. 2024].

Paranjape, B., Lundberg, S., Singh, S., Hajishirzi, H., Zettlemoyer, L. and Ribeiro, M.T., (2023) ART: Automatic multi-step reasoning and tool-use for large language models.

Pascale Fung and Chi-Shun, (2003) *Combining Optimal Clustering and Hidden Markov Models for Extractive Summarization*.

Patel, D., Shah, S. and Chhinkaniwala, H., (2019) Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. *Expert Systems with Applications*, 134, pp.167–177.

Paulus, R., Xiong, C. and Socher, R., (2017) A Deep Reinforced Model for Abstractive Summarization. [online] Available at: http://arxiv.org/abs/1705.04304.

Prithwiraj Bhattacharjee, (2021) *Bengali Abstractive News Summarization (BANS): A Neural Attention Approach*. [online] Available at: https://github.com/Prithwiraj12/Bengali-Deep-News-Summarization.

Qin, W. and Luo, X., (2023) A Legal News Summarisation Model Based on RoBERTa, T5 and Dilated Gated CNN. Institute of Electrical and Electronics Engineers (IEEE), pp.889–897.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) *Language Models are Unsupervised Multitask Learners*. [online] Available at: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe [Accessed 2 Feb. 2024].

Radhakrishnan, P. and Senthil kumar, G., (2023) Machine Learning-Based Automatic Text Summarization Techniques. *SN Computer Science*, 46.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., (2019) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. [online] Available at: http://arxiv.org/abs/1910.10683.

Ramakrishna et al, (2006) *A System for Query-Specific Document Summarization* *. [online] Available at: www.cnn.com.

Rau, L.F., Jacobs, P.S. and Zernik, U., (1989) Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management*, 254, pp.419–428.

Regina, B. and Michael, E., (1997) Using Lexical Chains for Text Summarization. [online] Available at: https://academiccommons.columbia.edu/doi/10.7916/D85B09VZ [Accessed 2 Feb. 2024].

Rehman, T., Das, S., Sanyal, D.K. and Chattopadhyay, S., (2023) An Analysis of Abstractive Text Summarization Using Pre-trained Models. [online] Available at: http://arxiv.org/abs/2303.12796.

Rudra, K., Goyal, P., Ganguly, N., Imran, M. and Mitra, P., (2019) Summarizing Situational Tweets in Crisis Scenarios: An Extractive-Abstractive Approach. *IEEE Transactions on Computational Social Systems*, 65, pp.981–993.

Rush, A.M., Chopra, S. and Weston, J., (2015) A Neural Attention Model for Abstractive Sentence Summarization. [online] Available at: http://arxiv.org/abs/1509.00685.

Sahba, R., Ebadi, N., Jamshidi, M. and Rad, P., (2018) Automatic Text Summarization Using Customizable Fuzzy Features and Attention on the Context and Vocabulary. In: *World Automation Congress Proceedings*. IEEE Computer Society, pp.68–73.

Sahoo, D., Bhoi, A. and Balabantaray, R.C., (2018) Hybrid Approach to Abstractive Summarization. In: *Procedia Computer Science*. Elsevier B.V., pp.1228–1237.

Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S. and Chadha, A., (2024) A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications.

Salton, G. and Buckley, C., (1988) *TERM-WEIGHTING APPROACHES IN AUTOMATIC TEXT RETRIEVAL*. *Information Processing & Management*, .

Sanh, V., Debut, L., Chaumond, J. and Wolf, T., (n.d.) *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. [online] Available at: https://github.com/huggingface/transformers.

See, A., Liu, P.J. and Manning, C.D., (2017) Get To The Point: Summarization with Pointer-Generator Networks. [online] Available at: http://arxiv.org/abs/1704.04368.

Sepúlveda-Torres, R., Vicente, M., Saquete, E., Lloret, E. and Palomar, M., (2021) HeadlineStanceChecker: Exploiting summarization to detect headline disinformation. *Journal of Web Semantics*, 71.

Shaik, T.S., Bharath, J., Darapaneni, N., Patra, S., Vishal, S.A., Manthripragada, S., Kagita, A.K., Rao, M. and Paduri, A.R., (2023) A Study of Text Summarization in the Medical Domain using BERT and its Variants. In: *2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC 2023*. Institute of Electrical and Electronics Engineers Inc., pp.969–975.

Sharifi, B.P., Inouye, D.I. and Kalita, J.K., (2014) Summarization of twitter microblogs. *Computer Journal*, 573, pp.378–402.

Shree Akshaya, A.T., Shankaran, S., Thrupthi, H.M. and Mamatha, H.R., (2022) Natural Language Processing based Cross Lingual Summarization. In: *2022 6th International Conference on Trends in Electronics and Informatics, ICOEI 2022 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp.1825–1829.

Singh, R.K., Khetarpaul, S., Gorantla, R. and Allada, S.G., (2021) SHEG: summarization and headline generation of news articles using deep learning. *Neural Computing and Applications*, 338, pp.3251–3265.

Sophie, S.L. and Siva Sathya, S., (2022) Extractive-Abstractive Summarization Using Transformers: A Hybrid Approach Automatic text summarization (ATS) minimizes a lengthy text document into a condensed. *Journal of Pharmaceutical Negative Results ¦*, 13.

Sutskever, I., Vinyals, O. and Le V, Q., (2014) *Sequence to Sequence Learning with Neural Networks*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf [Accessed 2 Feb. 2024].

Takeshita, S., Green, T., Friedrich, N., Eckert, K. and Ponzetto, S.P., (2023) Cross-lingual extreme summarization of scholarly documents. *International Journal on Digital Libraries*.

Tanaka, H., Kinoshita, A., Kobayakawa, T., Kumano, T. and Kato, N., (2009a) *Syntax-Driven Sentence Revision for Broadcast News Summarization*.

Tanaka, H., Kinoshita, A., Kobayakawa, T., Kumano, T. and Kato, N., (2009b) *Syntax-Driven Sentence Revision for Broadcast News Summarization*.

Tang Jie, Yao Limin and Chen Dewei, (2009) *Multi-topic based Query-oriented Summarization* *. [online] Available at: http://www-nlpir.nist.gov/projects/duc/duc2005/tasks.html.

Taunk, D. and Varma, V., (2023) Summarizing Indian Languages using Multilingual Transformers based Models. [online] Available at: http://arxiv.org/abs/2303.16657.

Tomer, M. and Kumar, M., (2022) Multi-document extractive text summarization based on firefly algorithm. *Journal of King Saud University - Computer and Information Sciences*, 348, pp.6057–6065.

Tonmoy, S.M.T.I., Zaman, S.M.M., Jain, V., Rani, A., Rawte, V., Chadha, A. and Das, A., (2024) A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. and Lample, G., (2023) *LLaMA: Open and Efficient Foundation Language Models*. [online] Available at: https://github.com/facebookresearch/xformers.

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017a) *Attention Is All You Need*. [online] Available at: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf [Accessed 2 Feb. 2024].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., (2017b) Attention Is All You Need. [online] Available at: http://arxiv.org/abs/1706.03762.

Wang, D. and Li, T., (2012) Weighted consensus multi-document summarization. *Information Processing and Management*, 483, pp.513–523.

Wang, S., Zhao, X., Li, B., Ge, B. and Tang, D., (2017) Integrating Extractive and Abstractive Models for Long Text Summarization. In: *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*. Institute of Electrical and Electronics Engineers Inc., pp.305–312.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A. and Zhou, D., (2022) Self-Consistency Improves Chain of Thought Reasoning in Language Models.

Wang, Z., Zhang, H., Li, C.-L., Eisenschlos, J.M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C.-Y. and Pfister, T., (2024) Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. and Zhou, D., (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

Weston, J. and Sukhbaatar, S., (2023) System 2 Attention (is something you might need too).

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A. and Raffel, C., (2020) mT5: A massively multilingual pre-trained text-to-text transformer. [online] Available at: http://arxiv.org/abs/2010.11934.

Yang, L., Cai, X., Zhang, Y. and Shi, P., (2014) Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information Sciences*, 260, pp.37–50.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y. and Narasimhan, K., (2023a) Tree of Thoughts: Deliberate Problem Solving with Large Language Models.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y., (2022) ReAct: Synergizing Reasoning and Acting in Language Models.

Yao, Y., Li, Z. and Zhao, H., (2023b) Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning in Large Language Models.

Yousefi-Azar, M. and Hamey, L., (2017) Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68, pp.93–105.

Yu, W., Zhang, H., Pan, X., Ma, K., Wang, H. and Yu, D., (2023) Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models.

Zakraoui, J., Alja'am, J.M. and Salah, I., (2022) Domain-Specific Text Generation for Arabic Text Summarization. In: *Proceedings of the International Conference on Computer and Applications, ICCA 2022 - Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Zeng, W., Luo, W., Fidler, S. and Urtasun, R., (2016) Efficient Summarization with Read-Again and Copy Mechanism. [online] Available at: http://arxiv.org/abs/1611.03382.

Zeyad, A.M.A. and Biradar, A., (2023) A Hybrid Text Summarization Approach Using Neural Networks and Metaheuristic Algorithms. *International Journal of Safety and Security Engineering*, 133, pp.479–489.

Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., (2020) *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. [online] Available at: https://proceedings.mlr.press/v119/zhang20ae/zhang20ae.pdf [Accessed 2 Feb. 2024].

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., (2019) BERTScore: Evaluating Text Generation with BERT. [online] Available at: http://arxiv.org/abs/1904.09675.

Zhang, Z., Zhang, A., Li, M. and Smola, A., (2022) Automatic Chain of Thought Prompting in Large Language Models.

Zhao, C., Huang, T., Chowdhury, S.B.R., Chandrasekaran, M.K., McKeown, K. and Chaturvedi, S., (2022) Read Top News First: A Document Reordering Approach for Multi-Document News Summarization. [online] Available at: http://arxiv.org/abs/2203.10254.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y. and Wen, J.-R., (2023a) A Survey of Large Language Models.

Zhao, X., Li, M., Lu, W., Weber, C., Lee, J.H., Chu, K. and Wermter, S., (2023b) Enhancing Zero-Shot Chain-of-Thought Reasoning in Large Language Models through Logic.

Zheng, H.S., Mishra, S., Chen, X., Cheng, H.-T., Chi, E.H., Le, Q. V and Zhou, D., (2023) Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models.

Zheng, S., Li, Z., Wang, J., Qu, J., Liu, A., Zhao, L. and Chen, Z., (2022) Long-Document Cross-Lingual Summarization. [online] Available at: http://arxiv.org/abs/2212.00586.

Zhou, Y., Geng, X., Shen, T., Tao, C., Long, G., Lou, J.-G. and Shen, J., (2023) Thread of Thought Unraveling Chaotic Contexts.

ADDITIONAL REFERENCES

1. https://huggingface.co/datasets/cnn_dailymail
2. https://huggingface.co/docs/transformers/model_doc/bert
3. https://huggingface.co/docs/transformers/en/model_doc/t5
4. https://openai.com/gpt-4

APPENDIX A: RESEARCH PLAN

Gantt Chart (Project Management)

The tasks required to complete a Master's dissertation are listed in the below proposed Gantt chart and are distributed over a period of many weeks. Every task is symbolized by a row, and every week of the year is represented by a column.

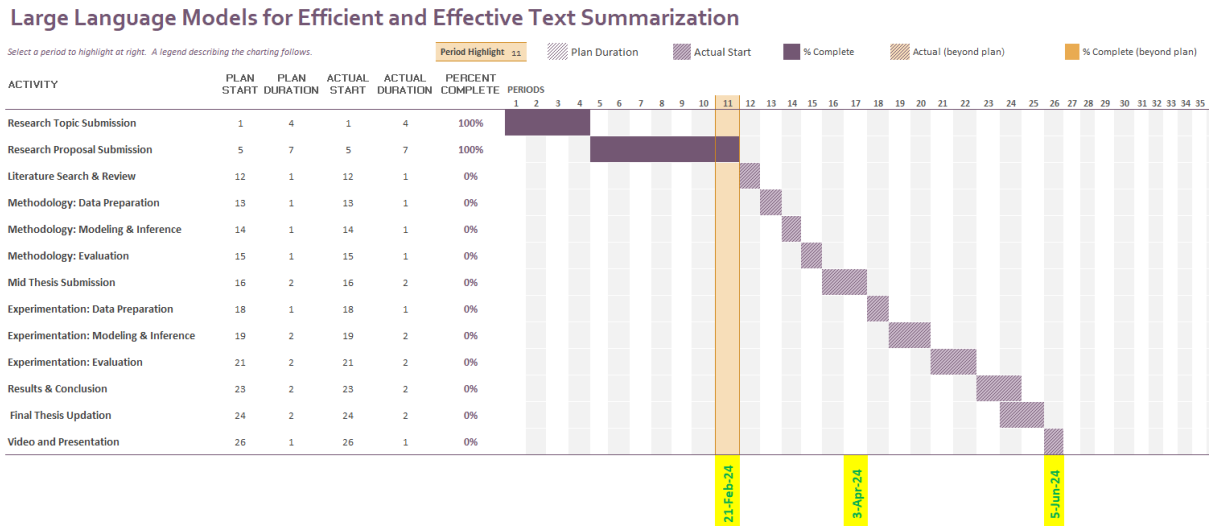

Figure Appendix A 1: Research Plan (Gantt Chart)

Note: 1 Row = 1 Task and 1 Column = 1 Week Period

9.2 Project Risk and Mitigation Plan

The potential risks associated with the research work along with appropriate mitigation plan is listed below:

Table Appendix A 1: Risk and Mitigation Plan

Potential Risk	Risk Mitigation Plan
Timelines are impacted when a candidate's health or personal concerns prevent them from conducting research work.	• Incorporate adequate buffer time in the project plan.

	• Request for extension from University and Upgrad.
Specialized graphics hardware, including GPUs or TPUs, is unavailable.	• Subscription to GPU infrastructure from Google Collab or Kaggle platform.

APPENDIX B: RESEARCH PROPOSAL

Large Language Models for Effective and Efficient Text Summarization

Harish Bekanahalli Nanjundappa

Research Proposal

February 2024

Abstract

News text summarization is the task of producing a short and accurate summary of a news article that captures the main information and highlights. It is a useful application for information retrieval, content analysis, and news aggregation. However, news text summarization poses several challenges, such as dealing with complex and diverse topics, preserving factual accuracy, and generating fluent and coherent summaries. In recent days, Large Language Models (LLMS) have demonstrated a great deal of promise for improving summarization methods. Our goal in this work is to investigate competing LLMS for text summarization, including BERT, T5 and GPT-4, from the perspectives of architecture, pre-training, fine-tuning, and assessment. The CNN/DailyMail news dataset will be used for this work, and the performance of the models will be evaluated against metrics like ROUGE and BLEU. We emphasize the benefits and drawbacks of each strategy as well as the problems that still need to be tackled in the field of LLM-based text summarization. We hope that our work will provide insight into the effectiveness of specific language models in text summarization and inspire new ideas and research in the field.

1. Background

News Text Summarization is the task of producing a concise and relevant summary of a news article. This task can help readers get the main information quickly, as well as assist journalists, editors and researchers in analyzing large amounts of news data. Text summarization techniques and models can be broadly classified into two categories: extractive and abstractive. Extractive methods select salient sentences or phrases from the input text and concatenate them to form a summary, On the other hand, sentences in source texts are rewritten using abstractive methods, which is more in line with how humans tackle the same issue. In theory, abstractive methods might produce summaries that seem more natural and are more efficient than extractive methods.

Extractive Summarization

Extractive methods have been dominant in text summarization research for a long time, due to their simplicity and effectiveness. Early extractive methods were based on statistical features, such as term frequency, inverse document frequency, position, or length, to score and rank sentences according to their importance. Earlier work by (Luhn H P, 1958) proposed word frequency and distribution to rank sentences significance, forming the basis of the summary. (Edmundson, 1969) extended Luhn's approach by considering additional factors like cue words, title words, and sentence location, making the method more comprehensive. (Kupiec et al., 1995) approach was more statistical and data-driven, relying on training feature weights using a corpus. (Ježek and Steinberger, 2004) proposed a generic text summarization technique in 2004 that leverages LSA to identify semantically significant sentences. As part of their work they also proposed newer evaluation methods, which measure similarity between an original document and its summary content, offering a new way to assess summary quality beyond just length. Lastly, (Mihalcea and Tarau, 2004) introduced TextRank in 2004, a graph-based ranking model for text processing. The basic idea is "voting" or "recommendation", where a vertex casting a vote for another increases its importance. They demonstrated that TextRank can be applied to summarizing single and multiple documents in any language.

Later methods incorporated more sophisticated linguistic features, such as discourse relations, lexical chains, or rhetorical structure, to improve the coherence and cohesion of the summaries. Author (Marcu, 1997) introduced an approach based on discourse structure, using a Rhetorical Structure Theory (RST) parser to identify summary units central to the document's claims. The structure is a binary tree assigning a status (nucleus or satellite) to each text span, with nucleus nodes deemed important for the summary. (Regina and Michael, 1997) introduced an approach leveraging lexical chains for text summarization and evaluation. This technique can produce high-quality summaries without requiring full semantic interpretation of the original text. In another research (Gunes Erkan, 2011) introduced LexRank, a graph-based method for multi-document extractive summarization. LexRank uses "degree centrality" in a graph where nodes represent sentences and edges represent sentence similarity. The authors also introduced "eigenvector centrality" or LexRank, a measure of a node's importance in a graph that considers the quality of links to a node, not just their quantity.

However, extractive methods have some inherent limitations, such as redundancy, informativeness, and readability. They cannot remove irrelevant or redundant information from the selected sentences, nor can they synthesize or compress information from multiple sources. They also cannot generate fluent and natural summaries that follow the conventions of human-written summaries. To overcome these limitations, researchers have explored abstractive methods, which aim to generate summaries that are closer to human-produced ones.

Abstractive Summarization

Abstractive methods have been challenging to develop, due to the difficulty of generating grammatical and coherent sentences that preserve the meaning of the input text. Early abstractive methods relied on manual or semi-automatic templates or rules to transform the input text into a summary, using techniques such as sentence fusion, sentence compression, or paraphrasing (Jing and Mckeown.,1999; Daumé et al., 2002; Barzilay and Lee, 2003). However, these methods were limited by the coverage and scalability of the templates or rules, and often required human intervention or domain knowledge.

Deep Learning Approaches

The recent breakthroughs in deep learning and neural networks have enabled significant progress in abstractive text summarization. Neural abstractive methods use sequence-to-sequence models, which consist of an encoder that encodes the input text into a vector representation, and a decoder that generates the summary from the vector representation, using attention mechanisms to concentrate on relevant parts of the input text (Sutskever et al., 2014; Chorowski and Bahdanau, 2015). Neural abstractive methods can generate fluent and readable summaries with less human effort and domain knowledge and can also incorporate copy or pointer mechanisms to deal with out-of-vocabulary words or rare entities (Rush et al., 2015; Chopra et al., 2016; See et al., 2017; Gū et al., 2019).

Despite the advances in neural abstractive methods, they still face several challenges, such as factual consistency, content selection, and evaluation. Neural abstractive methods can sometimes generate summaries that contain factual errors or inconsistencies with the input text, due to the lack of explicit reasoning or verification mechanisms (Cao et al., 2018; and Falke et al., 2019). Neural abstractive methods can also struggle with selecting the most salient and relevant information from the input text, especially when the input text is long or complex, or when the summarization goal is specific or query-focused (Nallapati et al., 2016b; Gehrmann et al., 2018). Moreover, neural abstractive methods are still difficult to evaluate, as the existing automatic metrics, such as ROUGE (Lin, 2004) or BLEU (Papineni et al., 2002), do not capture the semantic or pragmatic aspects of summarization quality, and the human evaluation is costly and subjective (Papineni et al., 2002; Lin, 2004; Liu and Lapata, 2019b).

Large Language Models

The use of large language models (LLMs) and pre-trained language models (PLMs) based on the Transformer architecture (Vaswani et al., 2017a) is one of the most recent developments in text summarization research. LLMs depends on large amounts of unlabeled text data to learn general language representations and generate natural language outputs BERT model (Devlin et al., 2019) Open GPT model (Radford et al., 2019), GPT 3 FSL (Brown et al., 2020).

(Paulus et al., 2017) proposed a novel approach that blends PLMs with reinforcement learning and graph neural networks to generate abstractive summaries that are more informative and consistent with the input text. RoBERTa (Liu et al., 2019) introduced a new pre-training objective for LLMs that encourages the model to generate concise and fluent summaries from long documents, without relying on any labeled summarization data.

LLMs can improve the performance and robustness of neural abstractive methods, as they can capture more semantic and syntactic information from the input text and generate more diverse and coherent summaries BERTSum (Liu and Lapata., 2019), PEGASUS (Zhang et al., 2020). LLMs can also enable zero-shot or few-shot learning for text summarization, where the model can generalize to new domains or tasks without fine-tuning or with minimal supervision UNILM(Dong et al., 2019) , T5, (Raffel et al., 2019) , BART (Lewis et al., 2019), BLOOM (Workshop et al., 2022), PaLM (Chowdhery et al., 2022), LaMDA(Thoppilan et al., 2022).

Google has released a new family of multimodal models Gemini (Gemini Team et al., 2023) that show impressive text, audio, video, and image interpreting skills. This model is among the first to attain Human Performance on 30 out of 32 state-of-the-art benchmarks.

However, LLMs are not without limitations and challenges. LLMs require a large amount of computational resources and memory to train and run, which poses ethical and environmental concerns and limits their accessibility and reproducibility (Schwartz et al., 2019, Strubell et al., 2019). LLMs can also suffer from factual inconsistency, content selection, and evaluation issues, as they are not explicitly trained for text summarization and may not align with the summarization objectives or expectations (Kryściński et al., 2019; Fabbri et al., 2020; Goyal and Durrett., 2020). Moreover, LLMs can generate summaries that are biased, misleading, or harmful, due to the potential biases or noises in the pre-training data or the generation process (Gehman et al., 2020; Bender et al., 2021). Finally, challenges exist in effectively controlling the length, style, and tone of the generated summaries, and adaptation of the model to different summarization scenarios and user preferences. Therefore, text summarization research based on LLMs is still an emerging and promising direction, with many research questions and challenges to be addressed.

Evaluation Metrics

One of the main challenges in text summarization research is how to evaluate the quality and usefulness of the generated summaries. Broadly. there are two approaches followed in the evaluation of text summaries - human approach and automated evaluation metrics.

Human Evaluation

Human evaluators are often considered superior due to their ability to assess aspects like coherence, conciseness, readability, and content. They can also compare two summaries and specify a preference. However, human evaluation has drawbacks such as time consumption, high costs, and inconsistency. For instance, the same judge might score the same summary differently at different times. These issues make a case for using automatic summarization metrics for evaluating generated text summaries.

Automated Evaluation Metrics

Human evaluation of text summarization is expensive, time consuming and may be biased and subjective. To alleviate these concerns a number of automated evaluation metrics are developed over past two decades.

BLEU Score (Papineni et al., 2002): An IBM-invented metric that compares the n-grams of machine-translated sentences to those of human-translated sentences. It counts the number of matches in a weighted fashion, with a higher match degree indicating a higher degree of similarity and a higher score. It doesn’t consider intelligibility and grammatical correctness.

ROUGE Score (Lin, 2004): Measures the overlap of n-grams in the generated summary and one or several human-constructed reference summaries. ROUGE-1, ROUGE-2, and ROUGE-L are the most commonly used versions, with ROUGE-L measuring the longest common sub-sequence. It’s popular due to its correlation with human judgments of summary quality.

METEOR (Banerjee and Lavie, 2005): This metrics takes into account both the precision and recall while evaluating a match. It was designed to fix some problems found in the BLEU metric and to correlate well with human judgment at the sentence or segment level.

BERTScore (Zhang et al., 2019): Leverages pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It computes precision, recall, and F1 measure, and has been shown to correlate with human judgment on sentence-level and system-level evaluation.

Text Summarization Datasets

The Automated Text Summarization algorithms require a large training dataset with ideal summaries (human annotated) to train the model. Many open source dataset are available for text summarizing; some of the well-known ones are described below.

CNN/DailyMail (Hermann et al., 2015; Nallapati et al., 2016b) An English-language dataset with over 300k unique news articles from CNN and the Daily Mail. It supports both extractive and abstractive summarization. Gigaword (Rush et al., 2015), it's used for headline-generation on a corpus of around 4 million English language articles.

WikiHow (Koupaee and Wang, 2018) it is one of the most widely used dataset for text summarization provided by NIST. It contains article and summary pairs extracted from an online knowledge base

LCTCS (Hu et al., 2015) More than 2 million authentic Chinese short text and its summary from Sino-Weibo are used to construct this dataset.

The Xsum dataset (Narayan et al., 2018) available for evaluating abstractive single-document summarization systems. It consists of news articles from BBC (2010 to 2017) with a one-sentence summary and covers a wide variety of domains. Multi News Dataset (Fabbri et al., 2019) is the first large-scale Multi document news dataset.

2. Related Work

Extractive Text Summarization

Narrain et al., 2023: Evaluated a hybrid approach for extractive summarization that combines pre-trained language models with graph-based methods. The hybrid approach achieved competitive results.

Harinatha et al.,2021: Compared multiple extractive summarization methods: LexRank, TextRank, and Luhn. LexRank and TextRank performed similarly and outperformed Luhn.

Lin et al., 2020: Proposed a method of knowledge distillation for extractive summarization. The method can reduce the model size and inference time without compromising the quality of the summaries.

Abstractive Text Summarization

Liu and Lapata, 2019a: Explored multi-document, abstractive summarization using Hierarchical transformers. The model leverages a pointer generator network and a coverage mechanism to deal with OOV words and repetition.

Paulus et al., 2017: Presented a new neural network model for abstractive summarization, which uses reinforcement learning to optimize the summary quality. The paper also introduces a large dataset for this task and shows that the model outperforms previous methods on CNN / Daily Mail and New York Times dataset.

Hybrid Text Summarization

Rahul et al., 2020: Reviewed recent advances in NLP and ML techniques for text summarization. The paper suggests that hybrid methods can achieve better performance and overcome the limitations of individual methods.

LLM based Text Summarization

Chhabra et al., 2024: Examined the problem of zero-shot abstractive summarization. They proposed a novel position-aware attention mechanism that can outperform previous methods and produce more informative and coherent summaries.

Rehman et al., 2023: Analysed four pre-trained language models for abstractive text summarization: BART, PEGASUS, ProphetNet, and T5. The models had issues with factual errors, semantic inconsistencies, and grammatical mistakes.

Ballout et al., 2023: Explored pre-trained language models (PLMs) like BERT, GPT, T5, BART, and PEGASUS for text summarization across different domains. PLMs can achieve SOTA results on cross-domain text summarization.

Munaf et al., 2023: Investigated how to use pre-trained language models like mBERT, mT5 for text summarization in low-resource language Urdu dataset. Their method can achieve competitive results with state-of-the-art models like BERT and T5.

Basyal and Sanghvi, 2023: Compared three large language models that can generate natural language instructions: MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT. They also discussed the ethical and social implications of using LLMs and PLMs for text summarization and instruction generation.

Pokale et al., 2023: Presented a novel approach to text summarization using GPT models. They introduced a new evaluation metric, ROUGE-GPT, that measures the quality of summaries based on the similarity of their hidden representations with the original texts.

Domain Related Summarization

Van Veen et al., 2023: Used BART and T5 as base models for clinical text summarization. The adapted models outperformed the base models and previous state-of-the-art models.

Pavlyshenko, 2023: Presented a novel approach to analyse financial news articles using a fine-tuned Llama 2 GPT model. The model outperformed other existing models on several metrics.

Umejiaku et al., 2022: Proposed an ensemble of BART and PEGASUS models with TextRank on the Covid19 dataset. The ensemble outperformed individual models.

3. Research Questions

The following queries are attempted to be addressed by this study:

1. The previous methods largely used GPT 2 and 3.x variants as base model. Does the recent GPT 4 version improve the performance of text summarization?

2. Can Prompt tuning improve the summarization capabilities of LLM?

3. Can a hybrid approach with LLMs improve the overall text summarization capabilities?

4. How does LLMs based on different transformer architecture perform on selected dataset for text summarization?

4. Aim and Objectives

This study attempts to investigate the potential of pre trained language models and identify the best performing approach and model for the task of news text summarization.

Objectives:

- To do an exhaustive analysis of the existing literature about the task of text summarization on news datasets.

- To explore various transformer based LLM architecture and short list few of the language models for the research.

- To explore the feasibility of LLMs and then create a technique for leveraging short listed LLMs to generate clear and meaningful summaries.

- To compare and contrast the performance of different language models against text summarization task.

5. Significance of the Study

Traditional approach of text summarization using Statistical, Machine Learning and Deep Learning has been studied extensively, LLM based approach is relatively new, actively researched and relatively under-explored. By contributing the code, benchmarks, and new research to the body of current literature, this endeavour attempts to close these gaps. Additionally, this article examines current advancements in query-based text summarization.

In terms of application, this work will help journalists and editors to quickly and accurately summarize large amounts of information from various sources, such as press releases, reports, interviews, and social media. This can save time and resources, as well as enhance the readability and relevance of the news articles. News publishers can focus on generating summaries for different audiences, purposes, and platforms, such as headlines, abstracts, bullet points, tweets, or newsletters.

6. Scope of the Study

The following defines the research work's scope:

- The research project must be finished around sixteen weeks after the research proposal is submitted.
- Hugging Face library's pre-trained language models will be used in the research work.
- The experimentation will be conducted using publicly available GPU such as Google-Collab platform.
- Given the resource restrictions only pre trained models will be used for research, retraining the language models with news dataset will be out of research scope.
- Research work only focus on automated performance metrics like ROGUE and manual Human evaluation of the generated summaries is out of this research scope.

7. Research Methodology

In this work, publicly available news dataset will be used to test the text summarizing and query-based summarization capabilities of selected language models.

7.1 Dataset Description

This research is based on CNN / Daily Mail news dataset (Hermann et al., 2015; Nallapati et al., 2016b) . This dataset contains ~300K news records and is widely used benchmark dataset in the field of natural language processing, specifically for text summarization tasks. Each

example in the dataset includes a news article and an associated human annotated abstractive summary. The articles cover a diverse range of topics, including politics, sports, entertainment, and international news. This diversity makes the dataset suitable for training and evaluating models on various domains.

7.2 Data Preparation

The raw data from the CNN / Daily Mail dataset needs to be processed before it can be used for training a text summarizer model. The main steps involved in data processing are as follows:

- Cleaning the text and the summary. This involves removing any unnecessary or noisy information, such as HTML tags, advertisements, images, captions, references, etc. The text and the summary should also be normalized, such as by converting all letters to lowercase, removing punctuation, expanding contractions, etc.

- Tokenizing the text and the summary. This involves splitting the text and the summary into smaller units, such as words, sub words, or characters, depending on the model architecture. The tokenization process can be done by using a library such as NLTK, spaCy, or Hugging Face Transformers, which provide various tokenizers for different languages and models. The tokenized text and summary should be stored as lists of tokens, one pair per line.

- Encoding the text and the summary. This involves converting the tokens into numerical values, such as indices, embeddings, or features, that can be fed into the model. The encoding process can be done by using a library such as Hugging Face Transformers, which provide various encoders for different models and vocabularies. The encoded text and summary should be stored as arrays of integers or floats, one pair per line.

- Padding and truncating the text and the summary. This involves adjusting the length of the text and the summary to a fixed size, such as by adding zeros or removing tokens, so that they can be batched together and processed efficiently by the model. The padding and truncating process can be done by using a library such as PyTorch or TensorFlow, which provide various functions for padding and truncating sequences. The padded and

truncated text and summary should be stored as arrays of integers or floats, one pair per line.

7.3 Algorithms & Techniques Description

7.3.1 Language Models

LLMs are a class of artificial intelligence models that are designed to understand and generate human-like text. They have gained significant attention due to their impressive performance in various natural language processing tasks, including text summarization.

Different types of architectures are available for LLMS like transformer based, Recurrent Neural Network based, Hierarchical Attention Based and Graph Neural Networks. In this research, various LLMs based on transformer architectures will be evaluated.

- BERT (Encoder only transformer architecture)
- GPT-4 (Decoder only transformer architecture)
- T5 (Encoder-Decoder transformer architecture)

These architectures and models represent different approaches to LLM-based text summarization, each with its strengths and weaknesses, depending on the specific requirements of the task at hand.

7.3.2 Few Shot Learning Techniques

Few-shot learning is a type of machine learning that aims to learn from a very small amount of data and generalize to new tasks. It is inspired by the human ability to quickly adapt to new situations with minimal supervision. Few-shot learning is especially useful for domains where data is scarce, expensive, or difficult to obtain, such as NLP, computer vision, and speech recognition. There are different variations of few-shot learning:

- One-Shot Learning: Each class contains only one example used to train the model.. This is an extreme form of model training where the trained model is expected to generalize from a single example.

- Few-Shot Learning (K-Shot): The model is trained on a small number (K) of examples per class, where K is typically a small integer.

- Zero-Shot Learning: In this method, the model is trained on a task for which it has never seen any instances. It depends on extrapolating from comparable assignments or courses.

Few Short Learning addresses the challenges posed by limited labelled data in text summarization tasks. By employing innovative techniques such as transfer learning, meta-learning, semi-supervised learning and data augmentation, it enables the development of robust text summarization models that can generalize well even with minimal supervision.

7.3.3 Prompt Engineering

In the context of text summarization, prompt engineering refers to the process of creating and constructing input queries, or prompts, that direct a language model to produce desired summaries. This method is frequently applied to transformer-based models or pre-trained language models such as GPT 4. Prompt engineering can assist in customizing the summary procedure and enhancing the summaries that are produced in terms of relevance, coherence, and informativeness.

For this research work, focus will be on few-shot learning and designing specific prompts for extractive and abstractive text summarization on the selected news dataset.

7.4 Prompt Implementation

By default, language models tend to rephrase the input text and perform abstractive summarization when requested to summarize the input text. With carefully designed prompt we can instruct the model to strictly perform extractive summarization.

Sample prompt for extractive summarization

- Input: {Text message to be summarized}

- Prompt: “From Input text select the key sentences and output the verbatim without any changes, paraphrasing or rephrasing”

Sample prompt for abstractive summarization

- Input: {Text message to be summarized}
- Prompt: “Summarize given text message”

7.5 Evaluation Metrics

Several criteria are to be used in the evaluation of the condensed text. Only evaluation using automatic metrics is the subject of this paper. The human evaluation is outside the purview of this project.

For automated evaluation, following metrics are proposed:

- BLEU - Measures the n-gram overlap between the generated text and the reference (ground truth) text. It calculates precision by comparing the number of overlapping n-grams in the summary with those in the reference data.
- ROUGE - is a set of metrics that includes F1 score, recall and precision, with a focus on content overlap. It evaluates the overlap of n-grams (words or sequences) between the summary generated and the reference text.

8. Required Resources

8.1 Hardware Requirements

For this research project, the following hardware specifications must be satisfied:

- A decent desktop or laptop computer with internet connectivity that can be used for browsing, creating documents, developing and running Python programs.
- GPU or TPU availability for deep learning model training and inference.

8.2 Software Requirements

List of software assets required for research work are listed below:

- Latest Web-browser (with plugins enabled)
- Python 3.9+ and associated libraries like Numpy, Scipy, Pandas, NLTK, Spacy
- VsCode editor or Jupyter Notebook

- Graphics drivers (like NVIDIA and compatible CUDA libraries)
- Deep Learning frameworks - PyTorch and TensorFlow
- Pre trained language models (from HuggingFace site)

8.3 Dataset Requirements

CNN / Daily Mail News dataset from the Hugging Face website

References

Banerjee, S. and Lavie, A., (2005) *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. [online] Available at: https://aclanthology.org/W05-0909.pdf [Accessed 3 Feb. 2024].

Barzilay, R. and Lee, L., (2003) *Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment*. [online] Available at: https://arxiv.org/abs/cs/0304006 [Accessed 2 Feb. 2024].

Basyal, L. and Sanghvi, M., (2023) Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models. [online] Available at: http://arxiv.org/abs/2310.10449.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020) Language Models are Few-Shot Learners. [online] Available at: http://arxiv.org/abs/2005.14165.

Cao, Z., Li, W., Wei, F. and Li, S., (2018) *Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization*. [online] Available at: https://aclanthology.org/P18-1015.pdf [Accessed 2 Feb. 2024].

Chhabra, A., Askari, H. and Mohapatra, P., (2024) Revisiting Zero-Shot Abstractive Summarization in the Era of Large Language Models from the Perspective of Position Bias. [online] Available at: http://arxiv.org/abs/2401.01989.

Chopra, S., Auli, M. and Rush, A.M., (2016) *Abstractive Sentence Summarization with Attentive Recurrent Neural Networks*. [online] Available at: https://arxiv.org/pdf/1509.00685.pdf [Accessed 2 Feb. 2024].

Chorowski, J. and Bahdanau, D., (2015) *Attention-Based Models for Speech Recognition*. [online] Available at:

https://proceedings.neurips.cc/paper_files/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf [Accessed 2 Feb. 2024].

Daumé, H., Knight, K., Langkilde-Geary, I., Marcu, D. and Yamada, K., (2002) *The Importance of Lexicalized Syntax Models for Natural Language Generation Tasks*. [online] Available at: https://aclanthology.org/W02-2102.pdf [Accessed 2 Feb. 2024].

Devlin, J., Chang, M.-W., Lee, K., Google, K.T. and Language, A.I., (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] Available at: https://aclanthology.org/N19-1423.pdf [Accessed 2 Feb. 2024].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W., (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. [online] Available at: https://arxiv.org/pdf/2005.14165.pdf [Accessed 2 Feb. 2024].

Edmundson, H.P., (1969) *New Methods in Automatic Extracting*. [online] Available at: https://dl.acm.org/doi/abs/10.1145/321510.321519 [Accessed 2 Feb. 2024].

Fabbri, A.R., Li, I., She, T., Li, S. and Radev, D.R., (2019) Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. [online] Available at: http://arxiv.org/abs/1906.01749.

Falke, T., Ribeiro, L.F.R., Ajie Utama, P., Dagan, I. and Gurevych, I., (2019) *Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference*. [online] Association for Computational Linguistics. Available at: https://aclanthology.org/P18-1015.pdf [Accessed 2 Feb. 2024].

Gehrmann, S., Deng, Y. and Rush, A.M., (2018) Bottom-Up Abstractive Summarization. [online] Available at: http://arxiv.org/abs/1808.10792.

Gemini Team., (2023) Gemini: A Family of Highly Capable Multimodal Models. [online] Available at: http://arxiv.org/abs/2312.11805.

Gū, J., Shavarani, H.S. and Sarkar, A., (2019) Pointer-based Fusion of Bilingual Lexicons into Neural Machine Translation. [online] Available at: http://arxiv.org/abs/1909.07907.

Hermann, K.M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M. and Blunsom, P., (2015) Teaching Machines to Read and Comprehend. [online] Available at: http://arxiv.org/abs/1506.03340.

Hu, B., Chen, Q. and Zhu, F., (2015) *LCSTS: A Large Scale Chinese Short Text Summarization Dataset*. [online] Association for Computational Linguistics. Available at: http://www.nist.gov/tac/2015/KBP/.

Ježek, K. and Steinberger, J., (2004) *Automatic Text Summarization (The state of the art 2007 and new challenges)*. [online] Available at:

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=818dfb5d509c0571152a175f72825dc6f94569ab [Accessed 2 Feb. 2024].

Jing, H. and Mckeown, K.R., (1999) *The Decomposition of Human-Written Summary Sentences*. [online] Available at: https://dl.acm.org/doi/pdf/10.1145/312624.312666 [Accessed 2 Feb. 2024].

Koupaee, M. and Wang, W.Y., (2018) WikiHow: A Large Scale Text Summarization Dataset. [online] Available at: http://arxiv.org/abs/1810.09305.

Kupiec, J., Pedersen, J. and Chen, F., (1995) *A Trainable Document Summarizer*. [online] Available at: https://dl.acm.org/doi/pdf/10.1145/215206.215333 [Accessed 2 Feb. 2024].

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. and Ai, F., (2019) *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. [online] Available at: https://arxiv.org/pdf/2211.05100.pdf [Accessed 2 Feb. 2024].

Lin, C.-Y., (2004) *ROUGE: A Package for Automatic Evaluation of Summaries*. [online] Available at: https://aclanthology.org/W04-1013.pdf [Accessed 2 Feb. 2024].

Liu, Y. and Lapata, M., (2019) Text Summarization with Pretrained Encoders. [online] Available at: http://arxiv.org/abs/1908.08345.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. [online] Available at: http://arxiv.org/abs/1907.11692.

Luhn H P, (1958) *The Automatic Creation of Literature Abstracts*. [online] Available at: https://ieeexplore.ieee.org/abstract/document/5392672 [Accessed 2 Feb. 2024].

Marcu, D., (1997) *The Rhetorical Parsing of Natural Language Texts*. [online] Available at: https://aclanthology.org/P97-1013.pdf [Accessed 2 Feb. 2024].

Mihalcea, R. and Tarau, P., (2004) *TextRank: Bringing Order into Texts*. [online] Available at: https://aclanthology.org/W04-3252.pdf [Accessed 2 Feb. 2024].

Nallapati, R., Zhou, B., santos, C.N. dos, Gulcehre, C. and Xiang, B., (2016) Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. [online] Available at: http://arxiv.org/abs/1602.06023.

Narayan, S., Cohen, S.B. and Lapata, M., (2018) Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. [online] Available at: http://arxiv.org/abs/1808.08745.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J., (2002) *BLEU: a Method for Automatic Evaluation of Machine Translation*. [online] Available at: https://aclanthology.org/P02-1040.pdf [Accessed 2 Feb. 2024].

Paulus, R., Xiong, C. and Socher, R., (2017) A Deep Reinforced Model for Abstractive Summarization. [online] Available at: http://arxiv.org/abs/1705.04304.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) *Language Models are Unsupervised Multitask Learners*. [online] Available at: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe [Accessed 2 Feb. 2024].

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., (2019) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. [online] Available at: http://arxiv.org/abs/1910.10683.

Regina, B. and Michael, E., (1997) Using Lexical Chains for Text Summarization. [online] Available at: https://academiccommons.columbia.edu/doi/10.7916/D85B09VZ [Accessed 2 Feb. 2024].

Rehman, T., Das, S., Sanyal, D.K. and Chattopadhyay, S., (2023) An Analysis of Abstractive Text Summarization Using Pre-trained Models. [online] Available at: http://arxiv.org/abs/2303.12796.

Rush, A.M., Chopra, S. and Weston, J., (2015) A Neural Attention Model for Abstractive Sentence Summarization. [online] Available at: http://arxiv.org/abs/1509.00685.

See, A., Liu, P.J. and Manning, C.D., (2017) Get To The Point: Summarization with Pointer-Generator Networks. [online] Available at: http://arxiv.org/abs/1704.04368.

Sutskever, I., Vinyals, O. and Le V, Q., (2014) *Sequence to Sequence Learning with Neural Networks*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf [Accessed 2 Feb. 2024].

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) *Attention Is All You Need*. [online] Available at: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf [Accessed 2 Feb. 2024].

Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., (2020) *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. [online] Available at: https://proceedings.mlr.press/v119/zhang20ae/zhang20ae.pdf [Accessed 2 Feb. 2024].

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., (2019) BERTScore: Evaluating Text Generation with BERT. [online] Available at: http://arxiv.org/abs/1904.09675.