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.

**Table of Contents**

[1. Background 4](#_Toc159173828)

[2. Related Work 9](#_Toc159173829)

[3. Research Questions 11](#_Toc159173830)

[4. Aim and Objectives 11](#_Toc159173831)

[5. Significance of the Study 12](#_Toc159173832)

[6. Scope of the Study 12](#_Toc159173833)

[7. Research Methodology 13](#_Toc159173834)

[7.1 Dataset Description 13](#_Toc159173835)

[7.2 Data Preparation 13](#_Toc159173836)

[7.3 Algorithms & Techniques Description 14](#_Toc159173837)

[7.3.1 Language Models 14](#_Toc159173838)

[7.3.2 Few-Shot Learning (FSL) Techniques 14](#_Toc159173839)

[7.3.3 Prompt Engineering 15](#_Toc159173840)

[7.4 Prompt Implementation 15](#_Toc159173841)

[7.5 Evaluation Metrics 16](#_Toc159173842)

[8. Required Resources 16](#_Toc159173843)

[8.1 Hardware Requirements 16](#_Toc159173844)

[8.2 Software Requirements 16](#_Toc159173845)

[8.3 Dataset Requirements 17](#_Toc159173846)

[9. Research Plan 17](#_Toc159173847)

[9.1 Gantt Chart (Project Management) 17](#_Toc159173848)

[9.2 Project Risk and Mitigation Plan 17](#_Toc159173849)

[References 18](#_Toc159173850)

# 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., 2016; 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, 2019).

**Large Language Models**

The use of large language models (LLMs) and pre-trained language models (PLMs) based on the Transformer architecture (Vaswani et al., 2017) 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., 2016) 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., 2016) . 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 (FSL) 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)

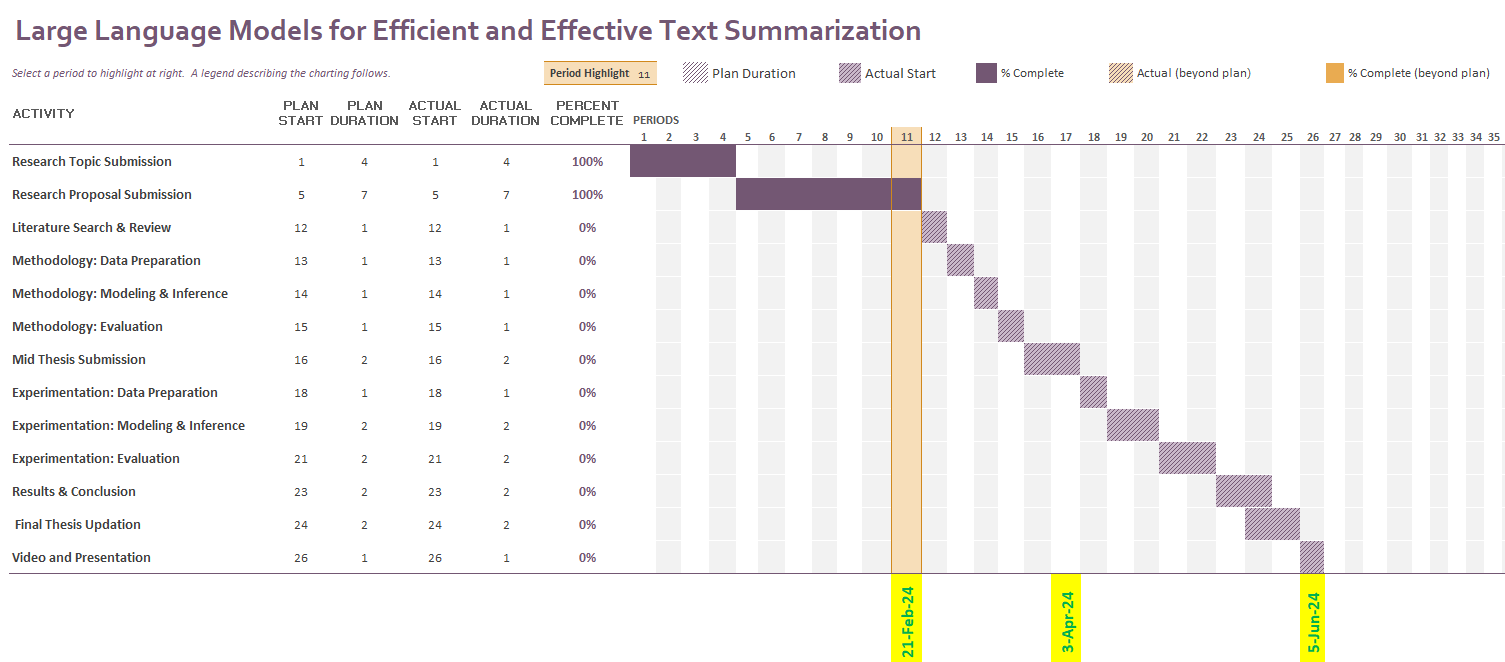
## 8.3 Dataset Requirements

CNN / Daily Mail News dataset from the Hugging Face website

# 9. Research Plan

## 9.1 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 9.1.1 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 9.2.1**

|  |  |
| --- | --- |
| **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. |

# 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.