SumBot: An enhanced multilingual Document Summarization using LLMs

MSc Research Project
MSc Data Analytics

Forename Surname
Student ID: x22208879

School of Computing
National College of Ireland

Supervisor: Syed Muhammad Raza Abidi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Anish Girish
Student ID:	x22208879
Programme:	MSc Data Analytics **Year:** September 2023
Module:	MSc Research project
Supervisor:	Syed Muhammad Raza Abidi
Submission Due Date:	12/08/2024
Project Title:	SumBot: An Enhanced Multilingual Document Summarisation using LLMs
Word Count:	6296 **Page Count:** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Anish Girish

Date: 11/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

SumBot: An enhanced multilingual document summarization using LLMs

Girish Anish
x22208879

Abstract

In a time where knowledge is available in excess, both written and spoken, summarising is an especially useful ability. Long texts are condensed into clear, comprehensive formats by summarization, which facilitates efficient communication and decision-making. This problem is addressed by automated document summarising, which uses Large Language Models (LLMs) and Natural Language Processing (NLP) to extract pertinent information from texts. Using extractive or abstractive approaches, this procedure identifies important words or concepts, preserving the main ideas of a document while eliminating unnecessary details. A unique hybrid framework called SumBot was created especially for the field of scientific literature to facilitate multi-document scientific summarization (MDSS). To produce high-quality summaries, this framework makes use of several Sentence Transformers and models from the T5 family. To adequately summarise entire material, the research focuses on analysing various kinds of LLMs and considering diverse document styles and languages. The study intends to improve automated summarization's accuracy and efficiency by analysing these models' performance, making it a useful tool for managing massive amounts of data in a variety of scenarios. This method helps better decision-making processes in a variety of disciplines and enhances information retrieval.

Keywords: Large Language Models, Hugging Face, ChatGPT, Unsupervised extractive summarization, Prompt Engineering

1. Introduction

The rapid expansion of online data calls for the development of artificial summarising technologies, which reduce large texts into succinct summaries to facilitate faster understanding. Summarization techniques have progressed from first statistical methods to contemporary methodologies such as Latent Semantic Analysis (LSA) and Non-Negative Matrix Factorization (NMF). In NLP, pre-trained models have the great performance. But, encoding scientific texts is a challenge due to their distinct style and specialized terminology (Sugimoto & Aizawa, 2004). Precise encoders are necessary for collecting scientific knowledge in its proper context, which is vital for efficient text summarization (Beltagy et al., 2019). Figure 1 is an example of document search engine.

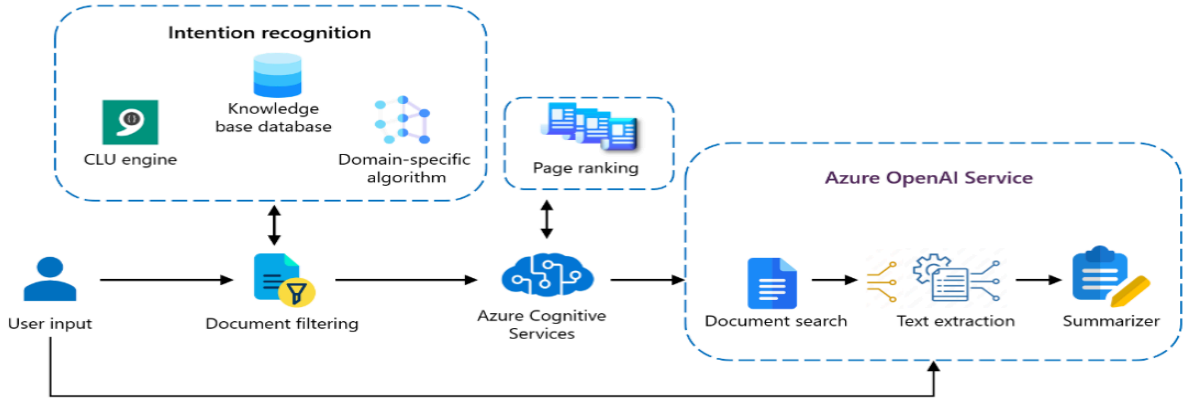

Figure 1: Document Search Engine

One significant obstacle is managing extensive datasets that surpass the token restrictions of models like as BERT, GPT-3, and T5. These models usually have token constraints of either 512 or 1024 tokens. This can only be done by the following chunking techniques such as divide, summarise and merge texts. This paper evaluates how well three models, T5, Bart Large, and Bart Large pretrained on Xsum dataset performed on the different datasets and aims at establishing which of the three models has the ability to retain most of the information and coherency. The contributions of the study are the demonstration of the effectiveness of fine-tuning and the comparison of the models for summarising when limitations on the number of tokens are considered. The goal is to enhance automatization of summing up and provide relevant information for further investigations.

The study conducts a comparison of three leading summary models: Google T5 fine-tuned on CNN, Facebook BART Large, and Facebook BART XSum. It means the goal is to find out which of the models under consideration makes the best summary, short, coherent and retains key information. The chosen model is also further tuned using chunked data regardless of the improvement in the summery abilities of the model. It is noteworthy that for increasing the effectiveness of the automatically generated summaries, it required enhancing the area of the automatic text summarization and studying strategies for fine-tuning and eliminating restrictions on the number of tokens. The desired outcome can thus be described as improving the processing and management of large volumes of textual information to speed up the search and facilitate the identification of relevant information in a range of contexts like academic work and information gathering through new services.

Thus, the study seeks to mitigate the token limitations of models like BERT and T5 by utilizing chunking to deal with extensive texts. The major purpose is to improve the approach of multiple document summarization in various fields. This involves basically evaluating a number of versions of the T5 model and comparing the results against the current models and benchmarks. The specific part of this paper is dedicated to the WikiHow dataset which has its specifics in the increased length of the text and the nature of “how to do it” sections. The aim is to supply brief and coherent resumes of the articles with special reference to the instructional aspect.

Some of the objectives are to enhance the processing rate of summaries and the efficiency of the preliminary content comprehension using such technologies as LLMs and serverless functions. The first papers in the domain of Multi-Document Summarization (MDSS) tended to use rather small data sets, and often relied on unsupervised extraction techniques (Duy et al., 2010; Jaidka et al., 2013). The literature review was conducted through the methods of statistical extractive summarization and citation-based surveys which were used in the study by the researchers (Gunes Erkan & Dragomir R Radev, 2004). That is why the prototype systems that were created ReWoS and Surveyor were to help in the creation of summaries. The new generation, however, experienced some problems in maintaining coherency and correctly gathering material (Duy et al., 2010; Jha et al., 2015) The basis for this work is formed by the limitations inherent to prior methods of the unsupervised learning approach, paying particular attention to the improvement of materials extracting with the aim of preserving and offering the relationships that are apt to promote comprehensive high-quality scientific summaries.

The aims of the research are to address the following research questions.

- What is the impact of using LLMs and short language models (SLMs) on the efficacy and accuracy of document summarization with comparison to traditional methods?
- What is the result of the extensive analysis of all the models?
- Which model provides the most value with respect to parameters, space, cost per usage and computing server?

This report will give a thorough analysis and solution to the current issue in the sections that follows. Section 2 goes with related works which focus on current research and relevant fields. Next will be the methodology and implementation in Section 3, including the data gathering methods, analytic strategies. The methods taken to solve the identified problem are outlined in the same section which presents the implementation approach. Section 4 has the analysis and results of the model discussed. Finally, section 5 concludes with the conclusion and the future work of the project.

2. Related Works

We know that there have been many approaches that have been made to utilize deep learning methods and especially very large sources of data for text summarization. (Koupaee & Wang, 2018) described an approach in their paper; creating an approach founded on a joint context-driven attention architecture needed to automate the summarizing of related work. Relative to five standard summarizing baselines and an average performant seq2seq summarizer, the authors' experimental results indicate that this strategy is far superior.

(Chen et al., 2023) also introduced another substantial work called Relation-aware Related Work Generator (RRG). While it was possible with this model to obtain abstractive summaries with the help of a Transformer-based architecture, it was not possible to obtain

highly informative, semantically remarked summaries. The Multi-XScience dataset that was released by (Lu et al., 2020) has many strong baselines that played a very important role in the MDSS study.

The function of PRIMERA by (Xiao et al., 2021) is to extract data from multiple documents, for which it is necessary to collect data, which is critical in the process of summarizing several documents. However, in Multi-XScience dataset it performed even poorer than the baselines. Nevertheless, using the extract-abstract architecture of KGSum (Wang et al., 2022). Thus, it can be concluded that both the extractor and abstractor, used in a single efficient approach, would be suitable for the MDSS challenge

2.1 Recent development in Abstractive Text Summarization (ATS) using LLMs

The incorporation of attention processes, which play a crucial role in the paper "Attention is All You Need" by (Vaswani et al., 2017), has greatly enhanced the efficiency of LLM-based summarization. These processes allow models to focus on certain sections of a document, hence improving the precision of capturing crucial points.

In addition, the hierarchical summarising technique described in the paper "Learning Hierarchical Document Representations for Abstractive Summarization" by (Liu & Lapata, 2019) allows for the generation of summaries at different degrees of detail, accommodating the differing information needs of users. (Jin et al., 2024) proposed the use of a process-oriented framework in summarising research. This framework focuses on practical aspects of real-world workflows, such as data preparation and model selection. The study "Fine-Tuning BART for Abstractive Reviews Summarization" conducted by (Yadav et al., 2022) focuses on improving the BART model for summarising reviews.

The research by (Chandra Challagundla & Reddy Peddavenkatagari, 2004) presents a novel framework for abstractive text summarization that combines structural, semantic, and neural-based approaches. By utilising Word2Vec embeddings and Bidirectional LSTM layers, this model effectively captures subtle semantic details to provide coherent summaries. Attention mechanisms guarantee the importance of information, whereas Word2Vec embeddings driven by Gensim improve the comprehension of meaning. The effectiveness of the design depends on its capacity to produce succinct summaries while understanding intricate semantic connections

The research of (Chen et al., 2023) focuses on overcoming the constraints of language models such as GPT-3/4 when it comes to checking the accuracy of summaries. The suggested technique enhances the quality of summaries by utilising knowledge graphs and structured semantics in conjunction with BART. This improvement is supported by ROUGE measurements and assessments conducted by humans. Furthermore, the results emphasise the importance of contextual data in improving LLM-based summarising techniques.

(Zhang et al., 2023) presents SummIt, an iterative framework for summarising that improves the accuracy of summaries and increases user pleasure by extracting relevant knowledge. Human assessments confirm its effectiveness, demonstrating substantial

enhancements compared to standard approaches. A recent study (Liu & Lapata, 2019) has proposed a new method that combines abstractive and extractive summarising techniques utilising BERT. This approach has shown improved quality and coherence in the summaries, outperforming the baseline models in terms of ROUGE scores and the handling of non-vocabulary terms. The research of (Thirunavukarasu et al., 2024) introduces an improved framework for summarising that utilises DistilBERT, T5, and GPT-based approaches.

2.2 Recent developments in multilingual text summarization

The use of Large Language Models (LLMs) enhances the capabilities of multilingual document summarization, providing a wide range of advantages. (Yadav et al., 2022) have highlighted the use of multilingual LLMs, which allow for summarising in different languages. These models improve communication across language boundaries and make it easier to create and translate material on a large scale. Nevertheless, there are still difficulties in effectively communicating nuances of language, as pointed out by (Jin et al., 2024), which has led to continuous investigation into the integration of cultural context. Recent research has shown that zero-shot cross-lingual summarising is able to avoid the requirement for fine-tuning on specific language pairings. This highlights the significance of user assistance in achieving the best outcomes for multilingual summary jobs. In general, the progress made in multilingual LLMs shows potential for promoting inclusive communication and comprehension in varied language environments.

2.3 Research on LLMs that can be deployed serverless

Due to the high demand for LLMs, researchers are finding ways on how to optimize the consumption of resources in order to adapt it to serverless platforms. "TinyBERT: In the study of “Lightweight Question Answering and Summarization through Distilled BERT” by (Jiao et al., 2019), the authors describe a technique for reducing the sizes of such pre-trained LLMs such as BERT. This compact model, TinyBERT, performs quite well on the summarization tasks, and at the same time it uses orders of magnitude less computational resources, which makes it applicable to the serverless environment.

The protection of the actual intellectual property of this LLM model is significant as well. Secure enclaves are separate modes of operation of a processor wherein the normal operations are confined to the secure area of the processor. They offer a secure environment where one can execute scripts and save confidential information (Yang et al., 2024). In the case of the LLM, the model code and weights can be executed and kept within a protected zone when the application is being used in a serverless setting. Too many benefits include isolation and security backed by the hardware. Some of the techniques that are being researched in the current include homomorphic Encryption method where computations can be made on the encrypted data hence safe inference of LLM models can be made without having to reveal the architecture of the models. Secure multi-party computation (SMPC) is a technology by which several people can compute a function with the inputs of other individuals without leaking own input to others. This can be utilized to mask some of the training data during training of LLMs in a federated manner.

It also proposed ServerlessLLM that is a serverless inference system for Large Language Model with GPU servers capacities, The proposed system optimizes the way of loading checkpoints so that it reduces the frequency of distant checkpoint downloads. new to the paper (Narayan et al., 2018) as shown in Figure 2, the concept known as integrated localization support is presented. In this way, Storage and Memory, the major components of computer, are used to maximize the potential of the company.

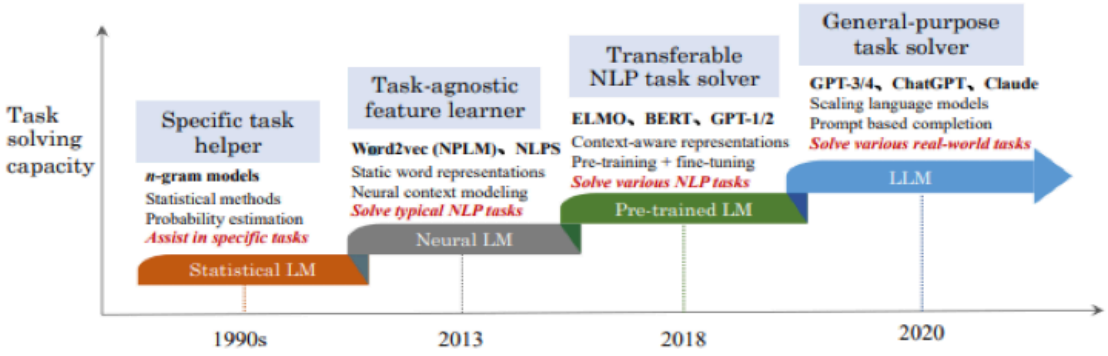

Figure 2: Four distinct LMs

2.4 Applications in the Real World and Case Studies

Researching possible impacts of using LLMs serverlessly for document summarization can benefit from case studies and empirical investigations. To validate the applicability and usefulness of such installations of serverless LLMs, researchers can measure performance metrics, users' satisfaction and costs on real-life settings. Security measures should be implemented when using LLMs on serverless systems so that sensitive data is not leaked and unauthorized attempts are prevented (Vaswani et al., 2017). These can be the topic of research proposals that can cover the methods for implementing access control measures, securing data transfer and ensuring compliance with data privacy laws such as GDPR and HIPAA.

3 Research Methodology and Implementation

3.1 Datasets

- CNN/DailyMail

A widely used benchmark dataset (Moritz et al., 2015) in the field of natural language processing (NLP) is the CNN/Daily Mail dataset. Activities like reading comprehension and text summarising benefit greatly from it. It was initially released in 2015 by Hermann et al. as a large-scale dataset for training and evaluating machine learning algorithms. This dataset includes news articles sourced from CNN and Daily Mail along with summaries written by humans. Politics, sports, entertainment, and technology are just a few of the many topics covered in this anthology of articles. Table 1 shows the description of the CNN/Daily mail dataset.

Table 1: CNN/Daily Mail data description

	Total	Average Length of Text	Average Length of Summary	Max length of article	Max length of summary	Min length of text	Min length of summary
Train	28112	773	57	2378	676	59	9
Test	11490	788	54	2886	1974	10	4
Validation	13368	764	60	2146	1716	45	10

- XSum

With extreme summary in mind, the XSum (Narayan et al., 2018) dataset was created as a carefully selected corpus. The goal of this assignment is to condense the main points of the source materials into concise one-sentence summaries. Articles on a wide range of topics were culled from the BBC News website and assembled here.

- WikiHow dataset

A large collection of how-to articles extracted from the WikiHow website is called the WikiHow dataset (Koupaee & Wang, 2018). More than 230k articles covering a vast array of topics are contained inside it. Among the many topics covered in these articles are recipes, health tips, and do-it-yourself projects. Typically, articles will provide a series of steps or directions to help readers complete a specific task or process. Table 2 shows the description of the WikiHow dataset.

Table 2: WikiHow data description

	Total	Average word in article	Average length of summary	Max length of article	Max length of summary	Min words in article	Min words in summary
Train	35774	1530	115	27494	1249	131	4
Test	2000	1526	115	10322	830	98	18
Val	2000	1552	115	5563	491	138	12

3.2 Large Language Models (LLMs)

As in Figure 3 the advent of LLMs has been a game-changer in NLP, altering the way computers engage with and understand specific human languages. This paradigm shift is being propelled by the groundbreaking transformer design, which deviates from the traditional sequential models. The transformer used a powerful technique called self-

attention, in contrast to its forerunners that processed text word by word. LLMs are given the ability to capture long-range dependencies and contextual nuances by the utilisation of this mechanism, which enables them to analyse the links between words over a full sentence or chapter. In a traditional style, the reader follows the lead of a young child reading aloud, with the focus being on individual words rather than the whole phrase. Conversely, the LLM can act like a seasoned linguist thanks to self-attention; it can understand the semantic structure and how words interact with one another.

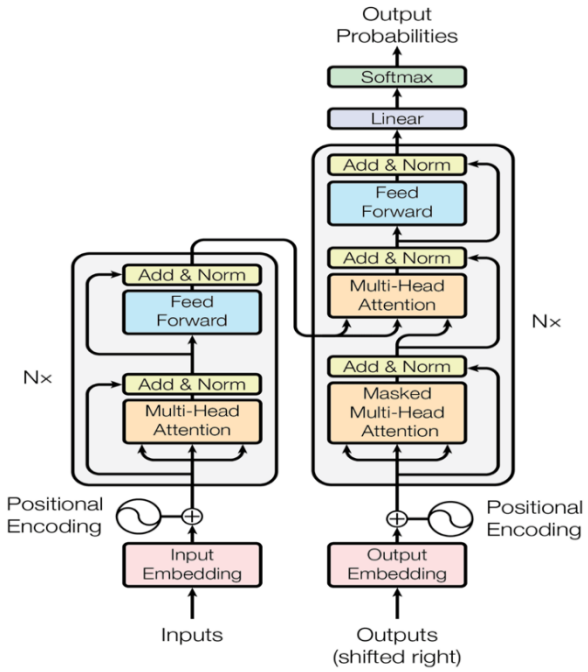

Figure 3: Transformation model

3.3 BART

The BART (Bidirectional and Auto-Regressive Transformers) architecture is a huge step forward in NLP since it is a denoising autoencoder built for pretraining sequence-to-sequence models. This architecture is known for its bidirectional encoder and left-to-right decoder as shown in Figure 4. The two-stage pretraining technique is a crucial part of BART's functionality. First, an entirely arbitrary noise production function is used to corrupt the text. The next step is to train a sequence-to-sequence model to reproduce the original text. An unparalleled degree of text corruption freedom is made available by this design, allowing for a wide range of modifications, including variations in text length. To achieve this, we thoroughly tested a wide variety of noise reduction strategies. Improving the model's reasoning abilities regarding overall sentence length and making longer-range changes to input text is the goal of this in-filling strategy. One way to achieve this is to replace strings of text of any length with a single mask token.

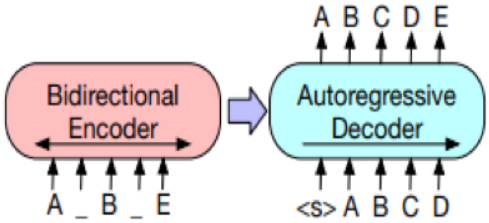

Figure 4: BART

3.3 T5 Model (Text-To-Text Transfer Transformer)

Naturally, Google Research's T5 approach has utterly transformed natural language processing. This paradigm introduces a new way of thinking about the age-old challenge of task classification with an emphasis on text-to-text issues. The use of this method allows for the consolidation of several tasks into one, such as translation, summarization, and answering questions. The famous and very efficient transformer design forms the basis of the T5 model, which is used for NLP. Originally designed for the Transformer, the T5 gadget makes use of encoder-decoder architecture. To transform the input text into continuous representations, the encoder employs a fully visible self-attention approach. As a result, long-range dependencies can be captured, as one token can take care of all the other tokens. Instead, the decoder uses causal (autoregressive) self-attention to generate the output text. Each token's generation is entirely reliant on tokens issued in the past to limit future information leaking. Use of residual connections and layer normalisation ensures effective information flow and ongoing training.

3.4 Data Processing

Data point filtering is an essential first step in data processing. One way to achieve this is to compare the document's length to a fifty-word threshold, while another way is to compare the summary's length to a five-word threshold. Assuming these prerequisites are satisfied, the data points are either considered or not. Iterations are performed over the partitions present within the dataset. Every time there's a split, a fresh list is made to contain the cleaned data points. Following that, iteratively processing the data points in the split and performing checks to guarantee that document and summary lengths meet the requirements are the next steps. The data point will be included to the cleansed list after the prerequisites are met. Finally, for each split, a dictionary containing the cleaned-up data is returned.

3.5 Implementation

Figure 5 shows the detailed Summarising part which is the SumBot proposed system for this problem.

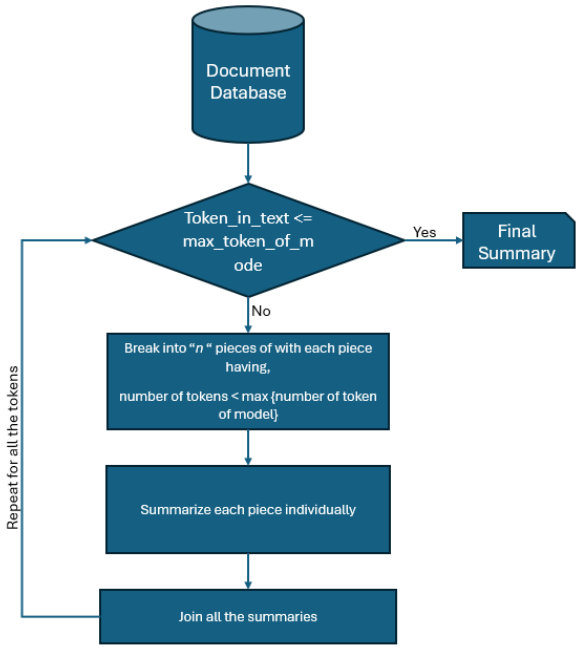

Figure 5: Flowchart of the summarization

3.5.1 Modelling

Stage 1: Document Extraction to create database in serverless environment

In this step as shown in Figure 6, to form a database, the content should be extracted. Since there will be different kind of data, it should be classified for each of the different kinds of the document. The Native documents can be extracted using the PDFMiner package in python. If the document contains images, tables and text, PDFminer will give the output for which all images or tables are present. While if the documents contains only scanned pages, then OCR needs to be done.

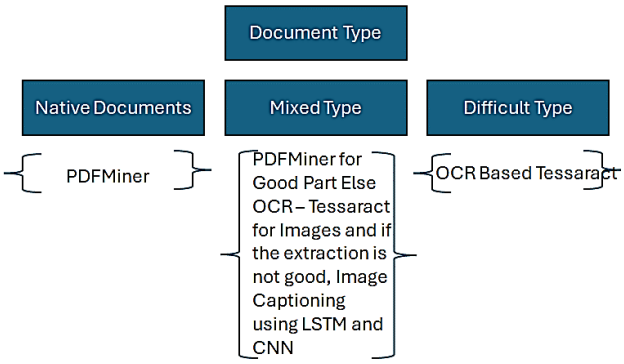

Figure 6: Document Classification technique

Stage 2: Document packaging

This process begins with extracting text from the document by identifying paragraphs by 'use of regex'. Then paragraphs will be packed together. Next the focus is to ensure the flow of the paragraph is smooth, preserving continuity. Finally it ends with adding caption to the images and extracting data from Excel file if data is present and organising the output.

Stage 3: Content Search

This stage contains of three steps as shown in Figure 7. First, the content will be stored in a databases like MySQL, NoSQL for proper search and effective retrieval. Second, question can be asked in any language contains images or tables. Python based OCR technique will be used to extract the content. But only one question can be asked at a time for better results. Finally, the top answers (ranging from 5 to 100) will be shown based on similarity measures such as Cosine, Jaccard and L1 and L2 Norms.

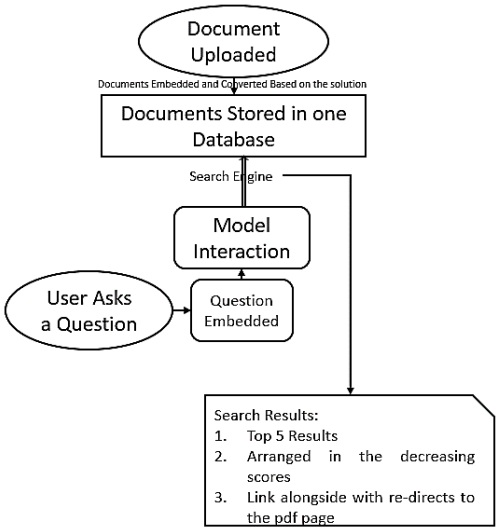

Figure 7: Outline of the Model generation

Stage 4: Summarization

Basically, this step is to generate the summary of the document using the hugging face. As the model should perform good in both server and serverless environments, it should be server independent. Figure 8 shows the building block of transformer module which acts like a base for LLMs.

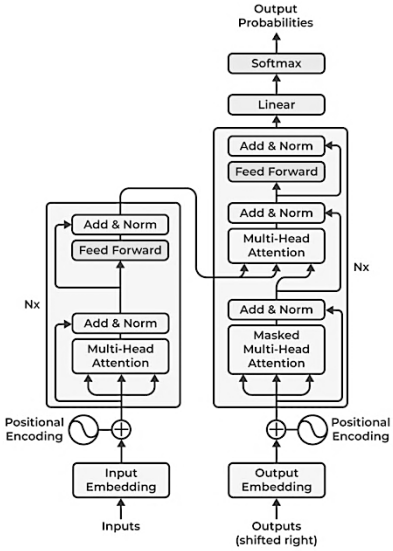

Figure 8: Building block of transformer module

3.6 Summary Generation

3.6.1 Using the models

For the goal of text summarization, three models that have previously been trained from Hugging Face are employed. The initialization of a tokenizer given by Hugging Face is the first stage in the process of converting the text into a format that the models can understand. For the best results, each model is taken from Hugging Face's model library and sent to the most suitable computing device, which might be a GPU or a CPU. Once the models and tokenizers are prepared, the summarising pipeline is built using Hugging Face's tools.

3.6.2 Chunking

Chunking is a very good technique in NLP for summarizing long texts that exceed the model's token limit. Tokens means dividing the text into smaller units. It starts with tokenization. Then text is divided into chunks where each of them has manageable number of tokens. Each chunk is summarized using a dedicated text generation pipeline with considering the length limit, beam search number and length penalties. Later all these are combined. The process is repeated if the combined summary still exceeds the token limit. This recursive method keeps the final summary within the length limit while maintaining its relevance and coherence.

3.6.3 Finetuning

A systematic approach is adopted to preprocess the data and configure the training parameters to fine-tune the BART-large models. The training and validation datasets are filtered to less than 1024 tokens, resulting in 8076 data points for training. This step ensures that the data is manageable within the model’s token limitations. The BART tokenizer is being used to prepare the input and target sequence during data preprocessing. The input articles will be tokenized to maximum length of 1024, while the summaries to 256 tokens. This uniform processing of data ensures that the model can handle the sequences efficiently. The output directory is being saved to './results'. For both training and evaluation, it will be configured to run for three epochs, with a per-device batch size of four.

3.7 Evaluation

- 3.7.1 ROUGE-1 (rouge1)

ROUGE-1 is a technique that figures out how many words, or unigrams, are shared between the reference summary and the created summary. To achieve this, it compares the reference summary with the created summary and counts how many unigrams are same. As a result, it can calculate measures like F1 score, recall, and precision. When testing a summarization model's ability to extract key phrases from source material, this variation is quite useful.

- 3.7.3 ROUGE-L

RougeL employs a different approach by finding the LCS between the reference summary and the generated summary. To accomplish this, it finds the longest string of terms that are

included in both summaries and uses this string to calculate the F1 score, recall, and precision. The ROUGE-L tool is quite useful for finding out how similar and coherent the generated summaries are to the reference summaries in terms of overall semantic similarity.

- 3.7.4 ROUGE-Lsum (rougeLSum)

ROUGE-Lsum enhances the evaluation process by considering the LCS for each generated summary with respect to the complete collection of reference summaries. Instead of comparing it to a single reference summary, it looks at how many references agree on something. This combined LCS is utilised as the foundation for computation of precision, recall, and F1 score. In cases where there are numerous reference summaries available, this version provides a more thorough examination. It sheds light on how well the summarising model performed across different types of reference literature.

4 Results and Discussion

4.1 The Pretrained Models

Our text summary experiments utilised three pre-trained models: Google T5 Base, Facebook BART Large that underwent fine-tuning on XSum, and another instance of Facebook BART Large. The output summaries generated by all the models were compared to the original summaries taken from the test datasets. To determine the effectiveness of each model in producing brief and precise summaries, their performance was evaluated.

4.1.1 GOOGLE t5 base

Based on the data and analysis of the ROUGE evaluation found in Table 3, it is evident that the study's use of an abstractive summarization approach yields a somewhat average performance. All three ROUGE measures show that the reference summaries and the generated summaries share a high degree of similarity, with mean scores ranging from around 0.3 to 0.4. Specifically, the ROUGE-1 test has the highest mean score, suggesting that the summaries retain a considerable amount of the original words used in the references. The generated summaries may not adhere strictly to the wording or sequence of the references, even though they convey the same substance, according to the lower mean scores for ROUGE-L.

Table 3: Performance Matrix for GOOGLE t5 base

	Mean	Median	Mode
Rouge1	0.400	0.400	0.139
RougeL	0.288	0.267	0.130
RougeLsum	0.322	0.326	0.132

Figure 9 shows the distribution of ROUGE-1 F1 score where Figure 10 shows the distribution of ROUGE-L F1 Score and Figure 11 shows the distribution of ROUGE-Lsum

F1 score.

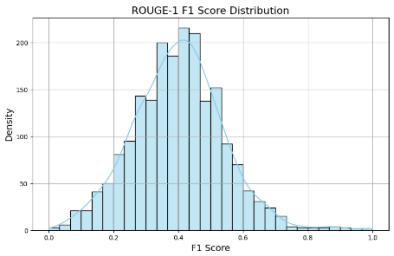

Figure 9

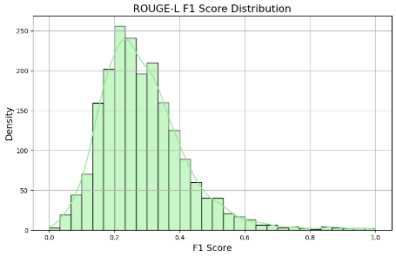

Figure 10

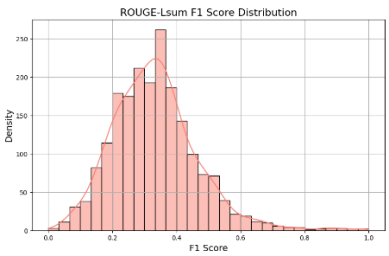

Figure 11

4.1.2 Facebook Bart Large

As shown in Table 4 the ROUGE- 1 matrix basically measures the unigram overlap between referred and generated summaries. In this case it shows positive with an average F1 score of 0.269 which means 26.9% of unigrams matched. The median score is 0.265 and standard deviation is 0.067 which suggests a wide range of scores. The mean ROUGE-L F1 score is 0.165 and median of 0.160 which shows that 16.5% of the longest common subsequence matched and has a standard deviation of 0.041which indicates higher consistency compared to ROUGE-1. ROUGE-LSum matric has a mean and median of 0.165 and 0.159 respectively and standard deviation of 0.041 which shows the model's performance sequence matching across different summaries.

Table 4: Performance Matrix for Facebook Bart Large

	Mean	Median	Standard Deviation
Rouge1	0.269	0.265	0.067
RougeL	0.165	0.160	0.041
RougeLsum	0.165	0.159	0.041

Figure 12 shows the distribution of ROUGE-1 F1 score where Figure 13 shows the distribution of ROUGE-L F1 Score and Figure 14 shows the distribution of ROUGE-Lsum F1 score.

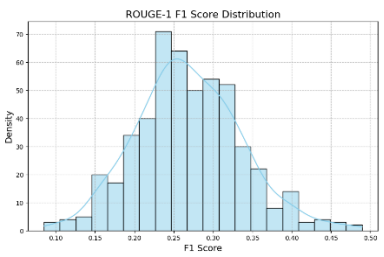

Figure 12

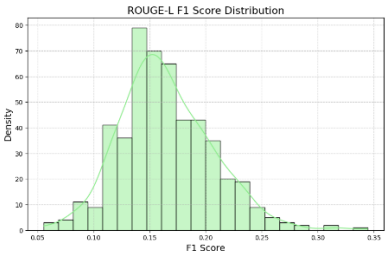

Figure 13

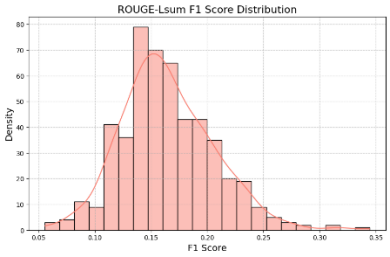

Figure 14

4.1.3 Facebook Bart Large Pretrained on XSum

With this model, positive results were discovered as shown in Table 5. ROUGE-1 F1 score is 0.288 which means 28.8% of the unigrams were identical to the reference summary. Consistent performance is shown with the average mean and median of 0.280. The standard deviation is 0.075. The median and mean ROUGE-L F1 score is 0.175 and 0.160 respectively which means data-based summaries made up 17.5% longest subsequence. The standard deviation is 0.051 which is very consistent. The mean and the standard deviation of ROUGE –LSum is same as ROUGE-L which shows the model's consistency where median is 0.169. ROUGE-LSum scores might be lower because abstractive summarization produces summaries which are more abstract and less textually aliened.

Table 5: Performance Matrix for Facebook Bart Large Pretrained on XSum

	Mean	**Median**	**Standard Deviation**
Rouge1	0.288	0.280	0.075
RougeL	0.175	0.160	0.051
RougeLsum	0.175	0.169	0.051

Figure 15 shows the distribution of ROUGE-1 F1 score where Figure 16 shows the distribution of ROUGE-L F1 Score and Figure 17 shows the distribution of ROUGE-Lsum F1 score.

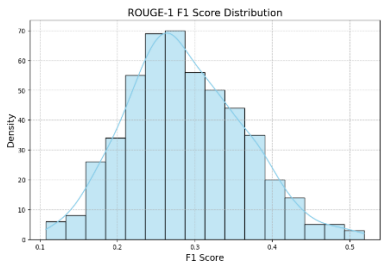

Figure 15

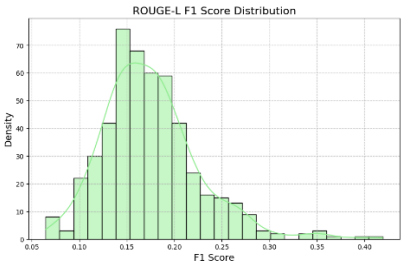

Figure 16

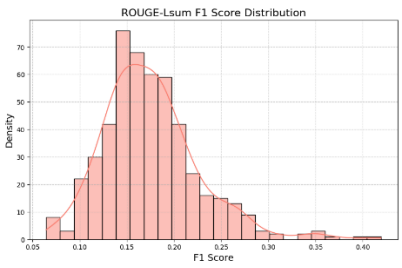

Figure 17

4.2 Finetuned BART Large

The BART model has an average ROUGE-1 F1 score of 0.461 and its median and standard deviation of 0.469 and 0.092 respectively after fine-tuning which can be seen in Table 6. This proves that the fine-tuned model achieved improved unigram overlap between the generated and reference summaries. The ROUGE-L F1 score average is 0.266. The median and standard deviation is 0.258 and 0.074 respectively. By this we can conclude that the higher

average score shows that sequence matching was more successful, even though the standard deviation was more, which means it is more variable. ROUGE-LSum F1 average score is 0.259. The median and standard deviation is 0.248 and 0.072 respectively. This model shows that it can produce better summaries with sequence coherence.

Table 6: Performance Matrix for Finetuned BART Large

	Mean	Median	Standard Deviation
Rouge1	0.461	0.460	0.092
RougeL	0.266	0.258	0.074
RougeLsum	0.259	0.248	0.072

Figure 18 shows the distribution of ROUGE-1 F1 score where Figure 19 shows the distribution of ROUGE-L F1 Score and Figure 20 shows the distribution of ROUGE-Lsum F1 score.

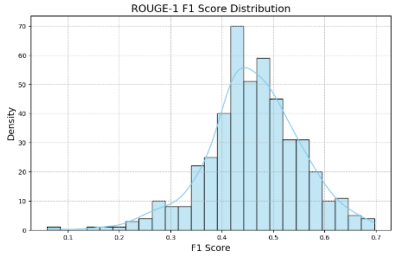

Figure 18

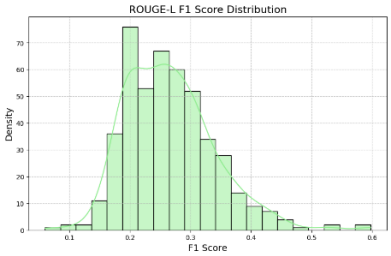

Figure 19

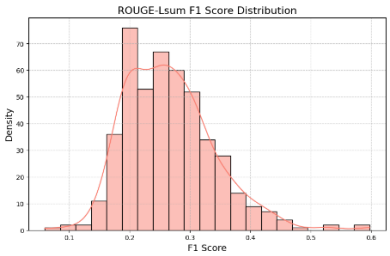

Figure 20

5 Conclusion and Future Work

In this study majorly chunking, tokenization was used to handle models with less token size. Here i compared three pre-trained models such as BART, LLM and T5 using CNN dataset. Main goal of this evaluation was to efficiently summarise article from WikiHow dataset. Since the articles were quite long and thorough while the summaries were more procedural in nature, this was an especially challenging problem to tackle. Extensive experiments demonstrated that the models differed significantly in terms of performance. The average ROUGE-1 score of 0.4 for the T5 model suggested that it was moderately effective at summarising. Here i got the average ROUGE-1 F1 score of 0.461 for CNN pre-trained BART which shows better performance than the other models. By seeing the results, we can say that the pretraining on domain-specific datasets improves summarization performance. The XSum dataset was used to fine-tune the BART, model might have absorbed better domain-specific details. The summarization will be more accurate as a result of this. It should be mentioned that chunking played a major role in circumventing the token size restrictions imposed by

large articles. This approach allows for faster processing of long articles without sacrificing the quality of the summaries, which contributes to the simpler nature of the models that produce summaries. There will always be some room for improvement, even though the pre-trained BART model on XSum dataset was demonstrated greatly. Models' performance can be enhanced further by investigating methods which optimize some articles in WikiHow dataset. Finally, this research clarifies the complex ways that lead to the best results between model architecture, and preprocessing procedures for optimal summarization outcomes. There will always be some room for improvement in the current technology, where researchers should always keep an eye on exploring new methods to increase performance.

5.1 Future Work

In order to make summarization models more efficient, it is suggested that future research look into more complex ways of fine-tuning. Some methods that could significantly increase performance include using more domain-specific pretraining data or trying out different fine-tuning approaches. The use of ensemble methods, which combine the predictions of many summarising models, can also greatly improve the accuracy and robustness of summarization. Additional refining is possible due to the investigation of approaches that can be utilised to adapt summarising techniques to the subtle characteristics of different article categories within datasets like WikiHow. By pursuing these paths, scientists might enhance text summarising models' efficiency and adaptability, leading to more accurate and contextually appropriate summaries in various fields and occupations.

6. References

Beltagy, I., Lo, K., & Cohan, A. (2019). *SciBERT: A Pretrained Language Model for Scientific Text*. http://arxiv.org/abs/1903.10676

Chandra Challagundla, B., & Reddy Peddavenkatagari, C. (n.d.). *Neural Sequence-to-Sequence Modeling with Attention by Leveraging Deep Learning Architectures for Enhanced Contextual Understanding in Abstractive Text Summarization*.

Chen, T., Wang, X., Yue, T., Bai, X., Le, C. X., & Wang, W. (2023). Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi-Source Transformers. *Applied Sciences (Switzerland)*, *13*(13). https://doi.org/10.3390/app13137753

Duy, C., Hoang, V., & Kan, M.-Y. (2010). *Towards Automated Related Work Summarization*. http://clair.si.umich.edu/clair/iopener/

Gunes Erkan, & Dragomir R Radev. (2004). *View of LexRank_ Graph-based Lexical Centrality as Salience in Text Summarization*.

Jaidka, K., Khoo, C. S. G., & Na, J. (2013). *Literature review writing: how information is selected and transformed*.

Jha, A. K., Huang, S. C. C., Sergushichev, A., Lampropoulou, V., Ivanova, Y., Loginicheva, E., Chmielewski, K., Stewart, K. M., Ashall, J., Everts, B., Pearce, E. J., Driggers, E. M., & Artyomov, M. N. (2015). Network integration of parallel metabolic and transcriptional data reveals metabolic modules that regulate macrophage polarization. *Immunity*, *42*(3), 419–430. https://doi.org/10.1016/j.immuni.2015.02.005

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2019). *TinyBERT: Distilling BERT for Natural Language Understanding*. http://arxiv.org/abs/1909.10351

Jin, H., Zhang, Y., Meng, D., Wang, J., & Tan, J. (2024). *A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods*. http://arxiv.org/abs/2403.02901

Koupaee, M., & Wang, W. Y. (2018). *WikiHow: A Large Scale Text Summarization Dataset*. http://arxiv.org/abs/1810.09305

Liu, Y., & Lapata, M. (2019). *Text Summarization with Pretrained Encoders*. http://arxiv.org/abs/1908.08345

Lu, Y., Dong, Y., & Charlin, L. (2020). *Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles*. http://arxiv.org/abs/2010.14235

Moritz, K., Tomáš, H. †, Kočisk´y, K., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P., & Deepmind, G. (n.d.). *Teaching Machines to Read and Comprehend*. http://www.github.com/deepmind/rc-data/

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

Sugimoto, K., & Aizawa, A. (n.d.). *Incorporating the Rhetoric of Scientific Language into Sentence Embeddings using Phrase-guided Distant Supervision and Metric Learning*. https://github.com/kaisugi/rhetorical_

Thirunavukarasu, A. J., Mahmood, S., Malem, A., Foster, W. P., Sanghera, R., Hassan, R., Zhou, S., Wong, S. W., Wong, Y. L., Chong, Y. J., Shakeel, A., Chang, Y.-H., Tan, B. K. J., Jain, N., Tan, T. F., Rauz, S., Ting, D. S. W., & Ting, D. S. J. (2024). Large language models approach expert-level clinical knowledge and reasoning in ophthalmology: A head-to-head cross-sectional study. *PLOS Digital Health*, *3*(4), e0000341. https://doi.org/10.1371/journal.pdig.0000341

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). *Attention Is All You Need*.

Wang, N., Liu, H., & Klabjan, D. (2022). *Large-Scale Multi-Document Summarization with Information Extraction and Compression*. http://arxiv.org/abs/2205.00548

Xiao, W., Beltagy, I., Carenini, G., & Cohan, A. (2021). *PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization*. http://arxiv.org/abs/2110.08499

Yadav, H., Nehal Patel, & Dishank Jani. (2022). *Fine-Tuning BART for Abstractive Reviews Summarization.* https://link.springer.com/chapter/10.1007/978-981-19-7346-8_32

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., & Hu, X. (2024). Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Transactions on Knowledge Discovery from Data*, *18*(6). https://doi.org/10.1145/3649506

Zhang, H., Liu, X., & Zhang, J. (2023). *SummIt: Iterative Text Summarization via ChatGPT*. http://arxiv.org/abs/2305.14835