National College of Ireland

Research Paper Summarization Using Text-To-Text Transfer Transformer (T5) Model

MSc Research Project
Data Analytics

Usama Hanif
Student ID: 22108831

School of Computing
National College of Ireland

Supervisor: Furqan Rustam

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Usama Hanif……..……

Student ID: ……x22108831……..……

Programme: ……Data Analytics……**Year:** ……2023…..……..

Module: ……Research Project……..……

Supervisor: ……Furqan Rustam……..……

Submission Due Date: ……18/09/2023……..……

Project Title: Research Paper Summarization Using Text-To-Text Transfer Transformer (T5) Model

Word Count: ……9830…… **Page Count**……26……..……..

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: ……**USAMA HANIF**……

Date: ……18/09/2023……..……

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):	

Research Paper Summarization Using Text-To-Text Transfer Transformer (T5) Model

Usama Hanif
X22108831

Abstract

There is a massive amount of research papers available online, and they keep increasing quickly but reading and understanding the sense of these long papers takes up a lot of time. This is where AI can lend a hand as it can help people understand these papers faster and pull out important information to create shorter summaries. In this study, the main focus was on a specific AI model called the Text-to-Text Transfer Transformer (T5) model. It was trained using a bunch of research papers and their summaries written by experts. The T5 model was fine-tuned so that it can become good at making brief yet meaningful summaries of these research papers. To evaluate the performance of the proposed approach, some measures like ROUGE and BLEU were used. This proposed approach scored well: it got 83% for ROUGE-1, 82% for ROUGE-2, 83% for ROUGE-3, and 47% for BLEU. These scores show how accurate and effective the approach is in summarizing the papers. The T5 model was also compared with other advanced models like BERT, GPT-2, and BART. The T5 model's summaries turned out to be more accurate in comparison. This study demonstrates that AI, especially the T5 model, can be a useful tool for quickly understanding complex research papers and creating helpful summaries. This could make it much easier for researchers and readers to get the main points from a lot of research without spending too much time.

1 Introduction

In today's world, research papers play a vital role in distributing knowledge, advancing various fields, and deepening the understanding of things. However, as the amount of information is increasing it is becoming difficult for researchers, scholars, and students to read and grasp every crucial key knowledge (Gupta *et al*. 2021). Research paper summarization is an essential tool that can help in summarizing and presenting the main points of lengthy papers in a concise and organized manner. By using the research paper summarization, readers can easily find the most important information without getting lost in lengthy and complex papers. By breaking down the barriers of verbosity and abstraction, summarization empowers the researchers to enhance their efficiency in going through large volumes of research, providing the means to stay up to date with the latest advancements and developing a deeper understanding of diverse subject matter.

Research paper summarization serves as a critical response to the challenge of information overload as millions of papers are published every year with more than 72 papers are published every day[1]. With such exponential growth in published research papers, accessing and comprehending each study in its entirety takes a large amount of time and leads to the oversight of crucial insights (Song *et al.*, 2019). Furthermore, for individuals seeking to conduct efficient literature reviews, summarization offers a time-saving approach, enabling them to identify relevant studies and key contributions quickly and access essential findings and methodologies without drowning in excessive details. Additionally, technologies like natural language processing and automated summarization techniques efficiently generate accurate summaries, reducing human effort and potential bias. Moreover, in time-sensitive domains relevant research summarization offers invaluable decision-making support, ensuring choices are grounded in the latest scientific evidence.

Deep learning is a revolutionary force within the domain of text summarization, as it is consisting of advanced algorithms and neural networks that can analyze a vast amount of text with efficiency and precision (Dong, 2018). The deep learning models especially which are based on transformers architecture (Vaswani *et al.*, 2017) can uncover the key details and generate summaries that capture the essence of the original text. Furthermore, deep learning is making it faster and better to find the key information in areas like news, work, and colleges where it is needed to understand things quickly and make choices effectively.

1.1 Research Question:

How can the T5 a deep learning model of the Transformer based architecture model help in the effective summarization of research papers?

1.2 Research Objectives:

Research paper summarization using the T5 model (Raffel *et al.*, 2020) serves as the critical response to the challenges posed by the time-consuming information overload. By using the advanced technologies of deep learning models like T5, accurate summaries can be generated efficiently. Therefore, the main objectives of this research are the following:

- Exploring the principle and architecture of the T5 model and comprehending its capabilities in processing large research papers.
- Investigating the existing text summarization approaches by reviewing the related literature thoroughly.
- Identify the factors that influence the summarization quality of the T5 model including the input parameters, data pre-processing, and fine-tuning strategies.
- Comparing the summarization performance of the T5 model against the other state-of-the-art deep learning languages models like BERT, BART, and GPT-2.

[1] Thousands of scientists publish a paper every five days (nature.com)

1.3 Research Contribution:

Current research is contributing to the field of text summarization by introducing the effectiveness of T5 (Text-To-Text Transfer Transformer) model when used specially to generate concise summaries of the research papers. This research is also demonstrating the ability of the T5 model to generate coherent summaries that capture the essence of the research paper's content. Furthermore, this research also contributes to the evaluation of summarization quality by using metrics like ROUGE and BLEU to assess the performance of the summaries generated by T5 with other state of the art models. This research will open new doors toward the development of practical applications which can summarize the research papers efficiently.

A detailed review of the related work is done in section 2 of this report. The overall approach for a suitable methodology that is used to complete this research is explained in section 3. Moreover, sections 4 and 5 discuss the design specification and implementation of the research project in detail. Section 6 includes the experiments and their evaluation that were carried out in this research to find the best model and the last section 7, contains the conclusion, limitations, and future work for the research.

2 Related Work

Every research work includes a literature review as it provides a thorough analysis of previous studies and provides a concise overview of the problem, ongoing research, and the relevant theories to be explored. In the past numerous studies have been conducted in the context of text summarization. Some of the most crucial papers are the following:

2.1 Background of Text Summarization

In the late 1950s, Luhn (1958) developed the first frequency-based method to summarize articles by using the sentence scoring feature to find the most relevant sentences and recommend that starting and concluding parts in each section can be useful to identify the subject. Moreover, Kupiec *et al.* (1995) developed a trainable approach to summarize the documents using the Naïve Bayes technique and compared the generated summaries with reference summaries written by experts. During the evaluation, they found out that their model was able to detect 84% of sentences from summaries written by professionals.
In their study, Neto *et al.* (2002) performed the automatic summarization of a collection of different documents about computers and hardware, etc. They applied the Naïve Bayes and C4.5 decision tree models in two experiments, where one experiment consists of both automatically produced summaries while the other consist of manually produced summaries. They find out that at a 20% compression rate, their models got a 43% precision rate. In another study, Steinberger *et al.* (2004) use the singular value decomposition (SVD) in text summarization to identify the important part of the text. They test their summarizer on the RCV1 dataset that contains the 800K documents and got a 0.765 score using a cosine evaluator.

However, they find out that their summarizer was very sensitive during the lemmatization process and their future work was to improve the summarizer.
Yeh *et al.* (2005) proposed the two text summarization approaches of modified corpus-based approach (MCBA) that used the genetic algorithm and latent semantic analysis based T.R.M approach (LSA + T.R.M). They applied these two approaches to summarize 100 political articles and got an average F-measure of 46%. They found out that applying both genetic algorithm and LSA text summarization can perform both keyword level and semantic level analysis.

2.2 Text Summarization Using NLP

Zhang *et al.* (2019) in their paper performed the abstractive text summarization using CNN based seq2seq model. Additionally, they find out that their model was performing efficiently when compared to the RNN-based seq2seq model. They applied the model on two datasets DUC corpus and GigaWord corpus. After applying the model to datasets, they got the Rouge-L score of 35% for the GigaWord corpus and 26% for the DUC corpus and thus conclude that their model was able to perform better than RNN and GAN models.
Adelia *et al.* (2019) in their study used the RNN-based Bidirectional Gated Recurrent Unit (BiGRU) model to summarize the Indonesian text. They used the 1000 Indonesian language journal documents in their experiment. They evaluate their model using two scenarios with different hidden units. In the first scenario, where they used more hidden states got the ROUGE – 2 score of 0.1199, and in the second scenario with less hidden states, they got the ROUGE – 2 score of 0.00550. Moreover, so according to the results, they conclude that their model was performing better with more hidden states because of linguistic factors.
The combination of CNN and LSTM was used by Song *et al.* (2019), to perform the text summarization to overcome the structure-related issues. They applied the model on two DailyMail and CNN news datasets. They performed different pre-processing techniques and got the ROUGE-1 and ROUGE-2 scores of 35% and 18% respectively. They conclude that their model was able to perform better on multi-sentence documents and was also performing better than the seq2seq model. In another research, Moravvej *et al.* (2021) performed the supervised summarization using conditional generative adversarial network (CGAN) on biomedical text. They developed a new loss function for the discriminator so that the model can perform better. To evaluate their model, they randomly selected 500 medical articles from the PubMed Central database. They got 44% ROUGE-1 and 27% ROUGE-2 scores. After evaluation, they conclude that they needed to improve the sentence coherence of generated summaries.
In a paper by Rahman and Siddiqui (2019), they developed a MAPCoL model that can automatically generate summaries of large text. They applied their model to CNN/DailyMail dataset and find out that it was working better than the traditional LSTM model. Moreover, they got the ROUGE-L score of 40% and conclude that their model was performing better than state-of-the-art models. However, their model was not efficient for generating long summaries.
As the RNN model is effective for summarization, Tomer *et al.* (2020) developed a hybrid model combining the bidirectional LSTM with the fuzzy logic rules. The proposed FLSTM

model was applied to DUC, CNN, and DailyMail datasets for summarization. They compared their model with other models like LSA, and RBM and calculate the ROUGE scores. Moreover, they find out that their model got the ROUGE –L score of 46% while other model's scores were less than 43%. They mentioned in their study that in the future the research can be extended by replacing the fuzzy logic with new models.

The authors in this research (Paulus *et al.,* 2017), combined the reinforcement learning model with the supervised word prediction model to summarize the news using the CNN, DailyMail, and New York Times datasets. Their proposed hybrid model got the ROUGE-1 score of 41%. Moreover, their model produces higher quality and is also evaluated by human professionals from which they conclude that their model was able to outperform the state-of-the-art models.

2.3 Text Summarization Using Transformers

Several research papers were also reviewed in which the models based on transformers architecture were used which are the following:

2.3.1 Text Summarization Using BERT

The study by Gupta *et al.* (2021) used the combination of LSA and BERT models on the news text documents to extract useful sentences. Additionally, they used the TF-IDF for keyword extraction and the BERT encoder for creating the embedding of sentences. To train their model they used a news dataset collected from Kaggle and split it into several sets according to the different number of documents. To evaluate the model, they used the ROUGE metrics and got the ROUGE-L of 36% percent when applying the model to 100 documents. They mention in their paper that summaries generated in machine language may achieve higher accuracy.

Iwasaki *et al. (*2019) performed the abstractive text summarization using the BERT model in the Japanese language. They applied the model to the Livedoor news dataset which contains 130,000 Japanese news articles. From the results, they conclude that their model was able to generate summaries by capturing the key points but repeating the sentences. Moreover, their model was unable to handle the unknown words. Elsaid *et al.* (2022) in their paper also face the same issues when summarizing the documents in the Arabic language.

In their paper, Kieuvongngam *et al.* (2020) performed the text summarization of Covid-19 medical research articles using a text-to-text approach. The authors used the pre-trained models of BERT and GPT-2 and applied them to the COVID-19 open research dataset. They evaluate their model using the ROUGE metrics and got the ROUGE-1 50%, ROUGE-2 21%, and ROUGE- L 48% scores.

Ramina *et al.* (2020) used the BERT model to generate the topic-level summary according to the input from the user. Additionally, their summarizer work in such a way that the keyword from the user will go to the model, and the relevant information will be searched through Wikipedia and finally get summarized by the BERT model to generate a topic-level summary. They train their model on CNN/DailyMail dataset. During the evaluation, they found that their ROUGE-L score was 38.76%. However, their model was limited because there were not enough reference summaries available and sometimes the generated summaries were getting out of context.

In this study by Bani-Almarjeh *et al.* (2023), they used several transformers-based models like mBERT, AraBERT, ARAGPT2, and AraT5 for Arabic summarization. They created their own dataset which includes almost 85 thousand high-quality text–summary pairs. They also used the BERT2BERT-based encoder-decoder architecture for fine-tuning the models. They used the ROUGE metric to evaluate their models and find out that the fine-tuned AraT5 achieves the best performance with a ROUGE-L score of 47%. However, their model was only trained for generating single-sentence summaries from news sources and was unable to generate multi-sentence summaries.

In their paper, Zhang *et al.* (2020) applied the PEGASUS transformer model to 12 summarization tasks like news, science, emails, etc. They used the 12 datasets like CNN/DailyMail and XSUM etc. They got the ROUGE-L score of 41% on CNN/DailyMail and 39% on the XSUM dataset. They conclude that their model was generating good results when exposed to the unseen summaries. Furthermore, in another study, Miller *et al.* (2019) performed the research to summarize the lectures. They used the RESTful Python-based service that utilizes the combination of both K-Means clustering and the BERT model to identify the sentences for summaries. Moreover, the main goal of their study was to provide the students with summarized lecture content. From the generated summaries they conclude that their BERT model was performing well but was not at par with the other models like TextRank and need further improvements.

2.3.2 Text Summarization Using T5

Raffel *et al. (*2020) in their research developed a T5 model based on text-to-text architecture for solving several NLP tasks. They trained their model on the large corpus of text data (C4). Additionally, they cleaned the dataset thoroughly so that the trained model can be utilized on several tasks related to the text. They test their T5-11B model with 11 billion parameters on several datasets whereas they got a ROUGE-L score of 41% and a BLEU score of 28% when used on CNN/DailyMail dataset.

Ay *et al.* (2023) performed the news headline summarization in the Turkish language. They applied the T5-base model to a dataset collected from news sources in Turkey. For better results, they convert the whole dataset into lowercase letters and also convert the Turkish language characters into Latin characters. They used the ROUGE metrics for evaluation and got the ROUGE-1 score of 69%, ROUGE-2 score of 66%, and ROUGE-L score of 75% and find out that their model was able to perform better than state-of-the-art models. In another paper, Garg *et al.* (2021) applied the T5 and BART to news articles for text summarization. They collected their dataset through web scraping from the Economic Times and Times of India websites. They performed the basic pre-processing on their dataset like removing the HTML tags. They evaluated their dataset using ROUGE-L and BLEU metrics and got 63% and 35% scores respectively.

In their research, Fendji *et al.* (2021) developed a WATS-SMS a T5-based model to perform the summarization on French web pages and convert them into SMS. They trained their model on an OrangeSum dataset that contain 25,000 Wikipedia pages. After evaluation, they conclude that their model performed better than other models with the ROUGE-L score of 77%.

Additionally, they find out that their model was generating incomplete summaries due to the limitation of characters' length. Moreover, Chaurasia *et al.* (2023) developed a hybrid model by combining the T5 transformer model. LSTM and RNN. They applied their T5LSTM-RNN model to biomedical articles to generate a concise summary. They also applied a base RNN model and got a ROUGE-L score of 23% while their hybrid model got a ROUGE-L score of 26%. However, they find out that their model failed to generate a high-level overview of the document.

2.4 Summary of Related Work

The reviewed literature provide the comprehensive history of approaches that were used to generate concise and informative summaries of the textual content. However, some limitations like sentence repetition, unknown words, and the absence of reference summaries were still the challenge in past studies. Additionally, after a thorough investigation, it was revealed that the T5 model was mostly applied on datasets like CNN/DailyMail to summarise news articles as can be seen in Table 1, and as the T5 model is an effective model, it was never used for summarizing the research papers. So according to these existing limitations and gaps, this research has been conducted to check the effectiveness of the T5 model against the research papers. The summary of the related work is shown in Table 1.

Table 1: Summary of Related Work

Reference	Technique	Data	Achieved Score
Kupiec *et al.*, 1995	Naïve Bayes	Random Documents	Human Evaluation
Neto *et al.*, 2002	Naïve Bayes & Decision Tree	Random Documents	43 % Precision Rate
Steinberger *et al.*, 2004	Singular Value Decomposition	RCV1 Dataset	0.765 Cosine Value
Yeh *et al.*, 2005	LSA + TRM	100 Political Articles	46 % F-measure value
Paulus *et al.*, 2017	Reinforcement Learning	CNN, DailyMail, and New York Times Dataset	41% ROUGE-1 score
Zhang *et al.*, 2019	CNN-based seq2seq model	DUC and GigaWord Corpus	ROUGE-L score of 35% and 26%
Adelia *et al.*, 2019	RNN based BiGRU	1000 Indonesian Journal Language Documents	ROUGE-2 of 0.1199
Song *et al.*, 2019	CNN and LSTM	DailyMail and CNN	35 % ROUGE-1 and 18% ROUGE-2
Rahman and Siddiqui 2019	MAPCoL Model	CNN and DailyMail	40% ROUGE-L
Tomer *et al.*, 2020	FLSTM	DUC, CNN, and DailyMail	46% ROUGE-L

Kieuvongngam *et al.*, 2020	BERT and GPT2	Covid-19 Medical Research Articles	50% ROUGE-1, 21% ROUGE-2 and 48% ROUGE-L
Ramina *et al.*, 2020	BERT	CNN and DailyMail	38.76% ROUGE-L
Zhang *et al.*, 2020	PEGASUS	CNN, DailyMail and XSUM	41% ROUGE-L
Raffel *et al.*, 2020	T5	C4	41% ROUGE-l and 28% BLEU
Moravvej *et al.*, 2021	CGAN	PubMed Central Database	44% ROUGE-1 and 27% ROUGE-2
Gupta *et al.*, 2021	LSA and BERT	100 Random Documents	36% ROUGE-L
Garg *et al.*, 2021	T5 and BART	News Articles	63% ROUGE-L and 35% BLEU
Bani-Almarjeh *et al.*, 2023	mBERT, AraBERT, AraGPT2 and AraT5	85K Arabic Documents	47% ROUGE-L
Ay *et al.*, 2023	T5	Turkish News Articles	69% ROUGE-1, 66% ROUGE-2, and 75% ROUGE-L
Chaurasia *et al.*, 2023	T5LSTM-RNN	Biomedical Articles	26% ROUGE-L

3 Research Methodology

This section presents the methodology which has been used to accomplish this research. In any research, selecting the appropriate data mining methodology is very important to ensure the smooth development of the model. Additionally, in this research, the steps of the Knowledge Discovery in Databases (KDD) approach have been used which begin with the collection of data and end with the evaluation of results. By utilizing the KDD approach in the first step the dataset consisting of research papers was collected and then in the second step the pre-processing was applied on the dataset to make it suitable for the T5 model. Furthermore, in the third step, data was transformed into the appropriate form in the context of this research the data was tokenized and embedded. After the data was prepared in the next step it was fed to the model for training and in the last step, the model's performance was evaluated to get the knowledge. Figure 1 illustrates the steps of the KDD approach.

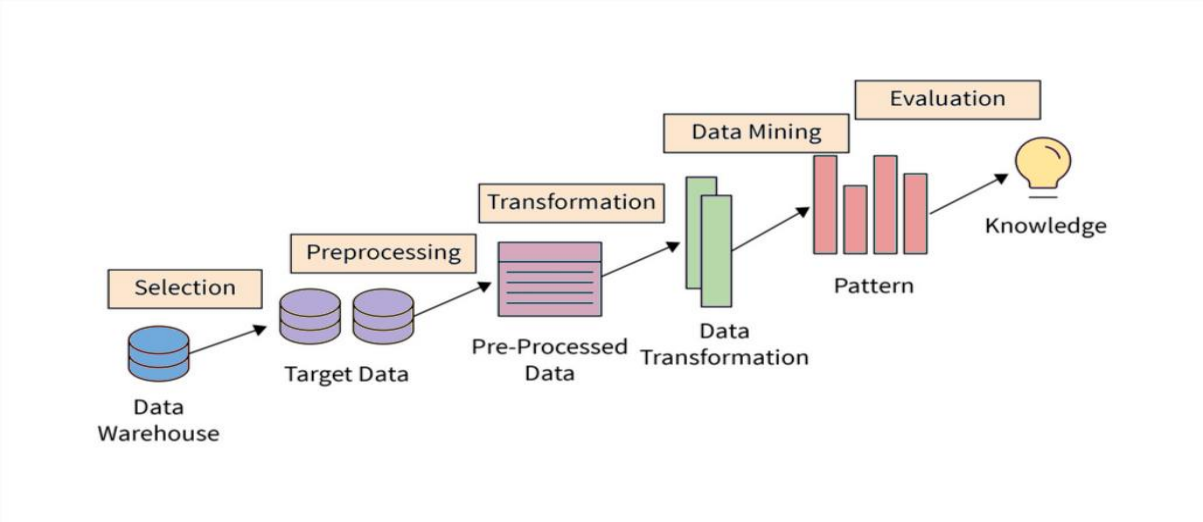

Figure 1: Knowledge Discovery in Databases (KDD) Methodology

3.1 Data Description:

Finding a suitable dataset was the first step of this research that contains research papers and their reference summaries. In this research, the Scisumm dataset created by Yasunaga *et al.*, (2017, 2019) was used that contains 1009 research papers related to the computational linguistics and NLP domain and also contains reference summaries that were written by professionals in the field. Moreover, as this research's main goal is to generate the summaries of the research papers, human-produced summaries were also necessary to evaluate the generated summaries. The dataset contains two columns: one is the Text column that contains the research papers and the second is the Summary column that contains the reference summaries of the papers. As can be seen from Figure 2, all the research papers are of variable length and mostly have 5000 words and their reference summaries are also variable in length and are mostly between 100 to 300 words.

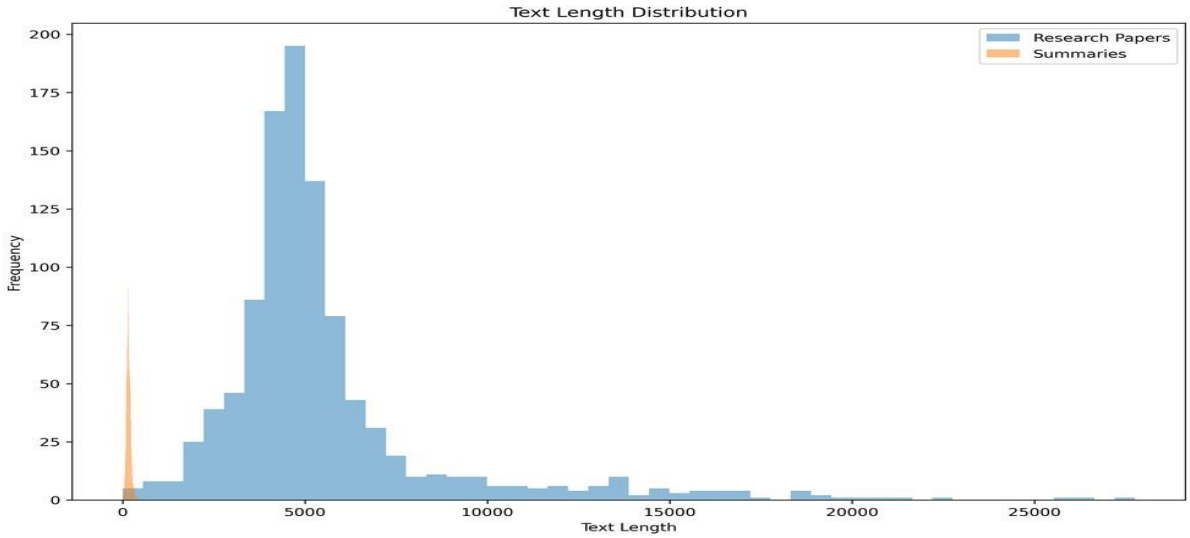

Figure 2: Dataset Length Distribution Chart

The dataset was also checked for missing and null values and no null or missing values were present in the dataset.

3.2 Data Pre-processing:

When the dataset was finalized, the next step was to pre-process the dataset into a form that was suitable to feed to the model for training. In most scenarios when pre-processing is done for text summarization, techniques like stemming, lemmatization, stop word removal, and special symbols removal is used. However, in the case of research paper summarization when the aforementioned techniques were applied the generated summaries lost their meaning and were impossible to read. As we can see in the following table:

Table 2: Comparison of Summaries Before and After Preprocessing

Before Preprocessing:
Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods\nThe task of paraphrasing is inherently familiar to speakers of all languages.\nMoreover, the task of automatically generating or extracting semantic equivalences for the various units of language — words, phrases, and sentences — is an important part of natural language processing (NLP) and is being increasingly employed to improve the performance of several NLP applications.\nIn this article, we attempt to conduct a comprehensive and application-independent survey of data-driven phrasal and sentential paraphrase generation methods, while also conveying an appreciation for the importance and potential use of paraphrases in the field of NLP research.\nRecent work done in manual and automatic construction of paraphrase corpora is also examined.\nWe also discuss the strategies used for evaluating paraphrase generation techniques and briefly explore some future trends in paraphrase generation.\nWe survey a variety of data-driven paraphrasing techniques, categorizing them based on the type of data that they use.\n
After Preprocessing:
generating phrasal sentential paraphrase : survey data-driven method task paraphrasing inherently familiar speaker language . moreover , task automatically generating extracting semantic equivalence various unit language — word , phrase , sentence — important part natural language processing (nlp) increasingly employed improve performance several nlp application . article , attempt conduct comprehensive application-independent survey data-driven phrasal sentential paraphrase generation method , also conveying appreciation importance potential use paraphrase field nlp research . recent work done manual automatic construction paraphrase corpus also examined . also discus strategy used evaluating paraphrase generation technique briefly explore future trend paraphrase generation . survey variety data driven paraphrasing technique , categorizing based type data use .

Additionally, stemming and lemmatization techniques reduce words to their root form, and in research papers, every word and its context is essential for accurately summarizing complex ideas and findings. Research papers also include equations, formulas, and other special symbols that hold crucial information, and removing these symbols can result in the loss of essential details. Furthermore, research papers often contain technical and domain-specific terms and when using the stop words removal these important terms can be discarded leading to less informative and less accurate summaries. However, as we can see in Table 2 that the sentences were losing meaning so the letters of research papers were only converted to

lowercase to reduce the vocabulary size because if the data is not converted to lowercase the words like “Data” and “data” will be treated as different words by the model (Ay *et al.*, 2023).

3.3 Data Transformation:

After the data was pre-processed the next step was to transform the data into the appropriate form for training the model. Tokenization is a process that is used in summarization tasks to break down the text into individual units called tokens (Grefenstette 1999). A specific Transformers Tokenizer was used to convert the inputs into the tokens and then the embedding was applied to convert the tokens into corresponding numerical IDs in the pre-trained vocabulary and thus transforming it into the format the transformer T5 model expects. Embedding is the process in which the words or tokens are converted into numerical vectors (Li *et al.*, 2018). The truncation and paddings were also used to keep all the data to the length the model expects. Furthermore, truncation is used for cutting the part of a sequence to the specified length while padding adds the special tokens at the start or end of the sequence to ensure the same length.

3.4 Model Building:

After the dataset was pre-processed and transformed, it was split into an 80% training set and a 20% test set. The T5 base version of the model was used in this research and was downloaded from the Huggingface website. The base T5 model contains 220 million parameters and 12 transformer layers with 768 hidden units in each layer. The T5 model is based on transformers architecture and consists of both encoder and decoder layers as can be seen in Figure 3. Moreover, it relies heavily on self-attention mechanisms to capture the long-term dependencies in input sequences (Vaswani *et al.*, 2017). In the attention layer, the softmax function is also applied to the weights to keep the distribution between 0 and 1. So in the context of lengthy research papers, this mechanism is very effective to generate meaningful summaries.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Tensorflow and Keras were used to fine-tune the T5 model for research paper summarization and the model was trained on Google Colab Pro, so that the large scale T5 model can be trained efficiently. The Adam optimizer was used due to its effectiveness in handling large neural networks like transformers and efficient training (Kingma *et al.*, 2014). Several hyperparameters like learning rate and batch sizes was changed to find the best fitted model as can be seen in section 6. At first the learning rate of 2e-5 was used but the model started overfitting so a smaller learning rate of 1e-5 was selected and the model started to improve (Malte *et al.*, 2019). Moreover, the reason to select such a small learning rate was to avoid a scenario where the model forgets the knowledge gained during pre-training. After selecting the

smaller learning rate the model further improved by changing the size of batches. At first the batch size of 8 was selected and the model was still overfitting to solve this issue the smaller batch size of 4 was selected because the dataset is also smaller in size. After finding the best hyperparameters the model was trained for 10 epochs to find the best-fitted model. The BERT model was also implemented with the same hyperparameters as T5 to generate summaries and for comparison purposes.

3.5 Model Evaluation

After training the model it was necessary to evaluate the performance of the model using different evaluation metrics. As can be seen in section 2 of this report, almost all the related studies used the ROUGE and BLEU metrics for evaluation. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of evaluation metrics that are mostly used for assessing the quality of generated summaries against human written reference summaries (Lin *et al.*, 2004). ROUGE measures the overlap of the sequence of words between generated summaries and reference summaries. The main ROUGE metrics are ROUGE-1, ROUGE-2, and ROUGE-L where ROUGE-1 and ROUGE-2 measure the n-grams while the ROUGE-L computes the longest common sequence (LCS) between the summaries including all words. Furthermore, higher ROUGE scores more than 50% provide the indication that the model is able to generate more good and meaningful summaries. ROUGE scores are usually the F1 scores which is the combination of both precision and recall as can be seen in the following equation:

$$R_{lcs} = \frac{LSC(X,Y)}{m}$$

$$P_{lcs} = \frac{LSC(X,Y)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$$

The BLEU (Bilingual Evaluation Understudy) is an evaluation metric that is mostly used to measure the quality of machine-generated texts (Papineni *et al.*, 2001). It evaluates the similarity between a reference summary and generated summary by comparing the sequences of words in both sentences. For many NLP tasks like text summarization which is comlex, a BLEU score higher than 30% is often considered good but it mostly depends on the complexity of the task. Furthermore, the BLEU combines various components to calculate the precision score of the sentence (S) which can be seen in the following equation:

$$BLEU_w(S) := BP(S).exp\left(\sum_{n=1}^{\infty} w\, lnp_n(S)\right)$$

Different experiments were performed by changing the batch size, learning rates, number of epochs, and models to find the best model for research paper summarization. The T5 model with a batch size of 4 was trained for 10 epochs and was the best model with the ROUGE-L score of 83% and BLEU score of 47% which means that the proposed model was able to generate the good meaningful summaires.

4 Design Specification

The T5 model has been used to generate summaries of the research papers in this paper. The T5 model is based on the transformer architecture of the encoder and decoder (Raffel *et al.*, 2020). The architecture of the T5 is made up of self-attention layers that enable it to support the variable length of inputs. Additionally, as the T5 model is based on transformer architecture, the transformer model contains the encoder and decoder layers with a multi-head attention layer and feed-forward network as shown in Figure 2. The encoder in the T5 model enables it to process the input text while the decoder generates the output text in text-to-text format. Moreover, in the T5 model, all NLP tasks are transformed into the text-to-text format to represent textual sequences, and the input is formulated with a specific task prefix as in this paper the prefix summarization was used and the model's goal is to generate the output according to the prefix. However, one major difference between the T5 and other transformer models is the use of reduced position embedding to compute the weights. The T5 model's key aspect is transfer learning because the T5 model was pre-trained on a large corpus of data and when trained on some specific data like research papers can achieve state-of-the-art performance by using its past training knowledge.

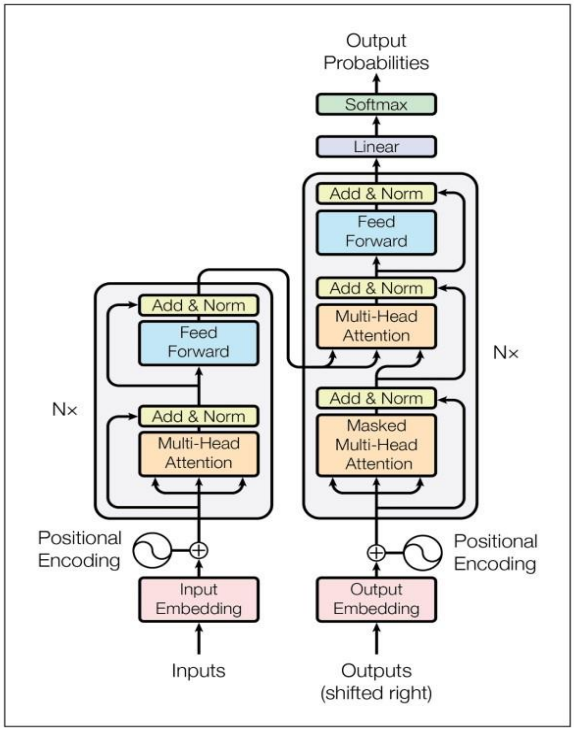

Figure 3: Transformer Model Architecture (Vaswani *et al.*, 2017)

Figure 4 illustrates the whole process that has been used in this research project. From Figure 4 we can see that the dataset consists of research papers split into train and test data sets and the T5 base model was selected for training, after the training of the T5 model, the results were evaluated using the performance metrics like ROUGE and BLEU and after that, the trained model was given research papers from the test set as input to generate the summary.

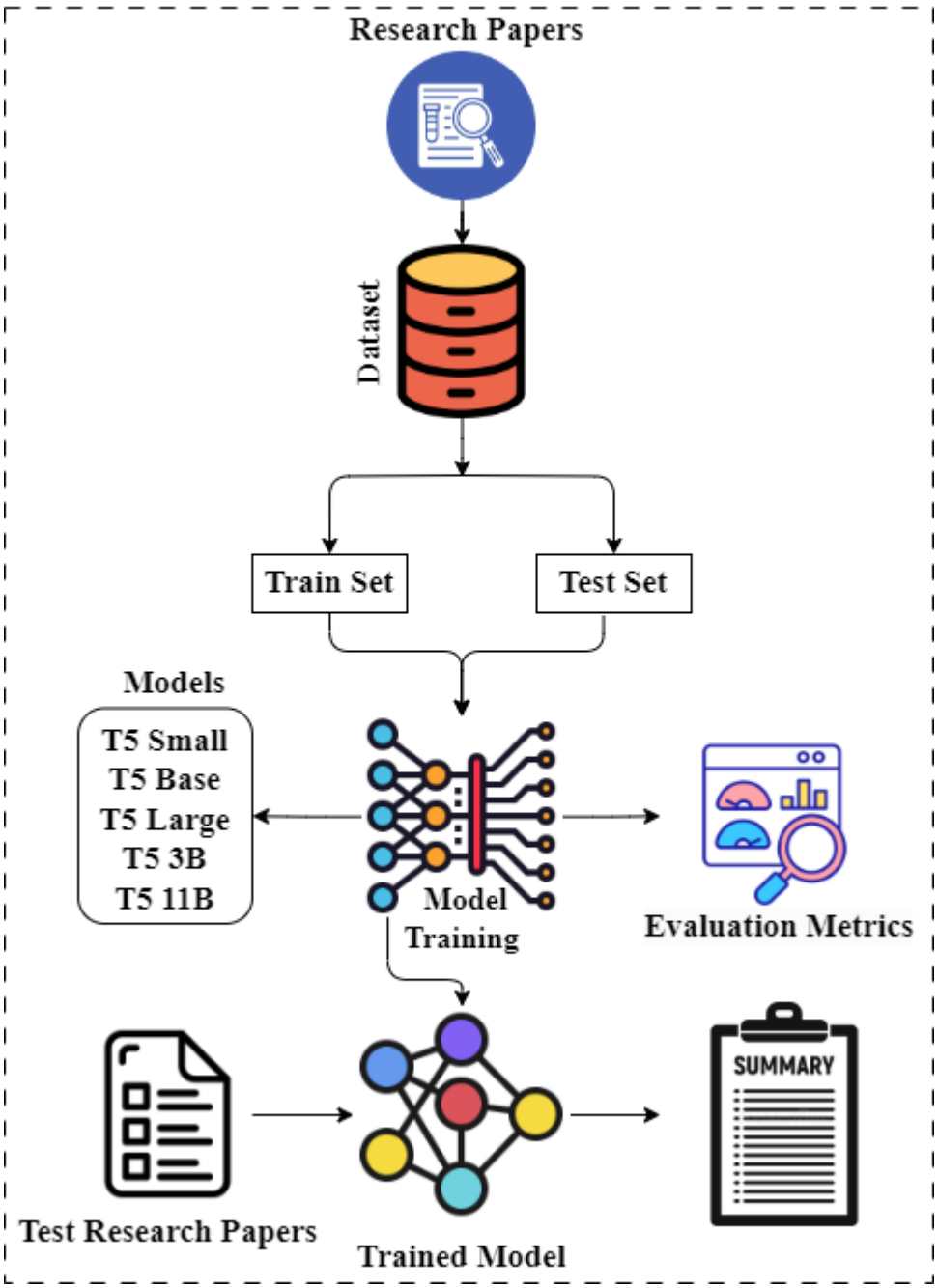

Figure 4: Research Paper Summarization Design Specification

5 Implementation

The implementation of this research project was done using Python programming language, TensorFlow, and Keras, and the platforms like Jupyter Notebook and Google Colab were used to train and evaluate the models. In this research, the T5 model was applied to the Scisumm dataset that contains the 1009 research papers with their human-generated reference summaries that were written by professionals. The dataset was first collected in the CSV format and the preprocessing was done on it.

During preprocessing the dataset was converted into lowercase letters so that the vocabulary size can be reduced if it is not done the model thinks that similar words like "science" and "Science" are different. As can be seen in Table 2, the techniques like lemmatization, stopword removal, and stemming were not applied to the dataset as it was resulting in summaries that were impossible to read and had no meaning. Moreover, the special symbols like % were also not removed because research papers often contain the evaluation scores in percentages and the mathematical equations are also often present in research papers, and by removing them the effectiveness of the generated summaries will be reduced. After the dataset was preprocessed it was uploaded to the Huggingface website so that it can be easily used in Google Colab.

The dataset was split into the train and test sets where 80% was used for training the model and 20% was used to test the trained model. The tokenization was applied to the preprocessed dataset to make it suitable for training the T5 model. In this research, the base version of the T5 model with 220 million parameters was used which was downloaded from the huggingface website. The AutoTokenizer transformer library was used to automatically load the most suitable tokenizer for the T5 base model. As both the input and output were the text sequences, the sequence-to-sequence language model was used to load the T5 model with seq2seq capabilities so that it can be used for text summarization. Moreover, a function was created to apply the tokenization on both research paper text and summary columns with the help of a tokenizer loaded by using AutoTokenizer. The truncation and padding were also applied so that all the text sequences have the same length. After that, a DataCollator for Seq2Seq class was loaded to format the tokenized data into the batches and create the necessary tensors suitable for feeding into the T5 model.

After the dataset was preprocessed and tokenized into a suitable form for the T5 model, the next step was to start the training of the model. The Adam optimizer was used because of its memory efficiency capability to handle large-scale models like T5. Furthermore, a small learning rate of value 1e-5 was used to prevent the overfitting of the model and to avoid a scenario where the model forgets the knowledge gained during pre-training. The batch size of 4 was used as the dataset only contains 1009 rows, the model was exposed to more varied data in each training step to improve the performance. Additionally, the other reason was the GPU memory constraints of the Google Colab because when the larger batch size was selected the resources of the platform were exhausted. After all, the T5 is a large model. During training, the model was iterated over the training data for 10 epochs, and the model was also evaluated on the test set to compute the validation loss.

After the model was trained, it was used to generate the summaries of the papers that were present in the test set. The evaluation of the model's performance was done using the ROUGE and BLEU metrics. Moreover, the trained model was saved on Google Drive for later use. The BERT model was also trained using the same hyperparameters that were used during the training of the T5 model. The BERT model's performance was also evaluated using the metrics like ROUGE and BLEU and it was also saved on Google Drive. Some pre-trained models like GPT2 and BART were also downloaded from the hugging face website to compare their performance with the trained T5 model

6 Evaluation

The evaluation is a crucial part of every research to measure the performance of the trained models. In this research paper, several experiments have been carried out to measure the performance of the T5 model for research paper summarization and also to compare the performance of different text summarization models against T5. As can be seen in section 2 of this report almost all the studies have used the ROUGE and BLEU metrics for evaluation. Several experiments were done to measure the quality of the T5 model for research paper summarization which includes changing its hyperparameters and comparing its performance with the other state-of-the-art language models which are the following:

6.1 Evaluating T5 Model with Batch Size of 8 and Learning Rate of 2e-5

In this first experiment, the T5 base model was applied to the preprocessed Scisumm dataset and trained for 50 epochs with a batch size of 8. As the T5 model is large so it was trained on Google Colab Pro with A100 GPU and during the training, the 33GB of GPU memory was used by the model. Additionally, this time slightly larger batch size was used to check how the model will perform but as we can in Figure 5 the model started to overfit.

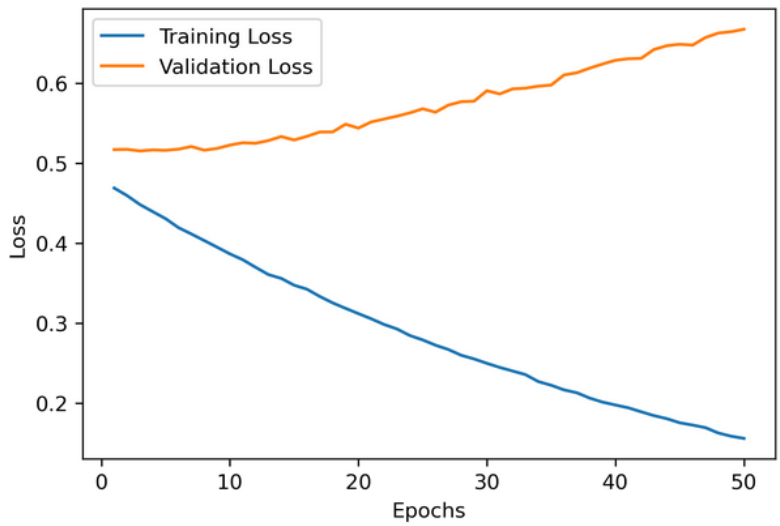

Figure 5: Loss Graph of the T5 model for 50 Epochs

Now from the above graph, it can be seen that the model was constantly learning with each epoch however it was not performing well when exposed to the unseen data. Furthermore, as

the loss was decreasing and validation loss was increasing it means the model was overfitting. The ROUGE and BLEU scores for this trained T5 model were calculated which are the following:

Table 3: Evaluation Scores of T5

Metrics	T5 Model
ROUGE-1	66%
ROUGE-2	60%
ROUGE-L	63%
BLEU	30%

6.2 Evaluating T5 Model with Batch Size of 4 and Learning Rate of 2e-5

In the second experiment, the T5 base model was applied to the preprocessed Scisumm dataset and trained for 10 epochs with a batch size of 4 and a learning rate of 2e-5. The model was trained on the Google Colab platform. As in the first experiment, the model was overfitting so it was necessary to make some changes to hyperparameters. Moreover, as the T5 model is a large-scale model whenever a larger batch size than 4 or 8 was chosen the resources were exhausted, so in this experiment, the smallest batch size was selected.

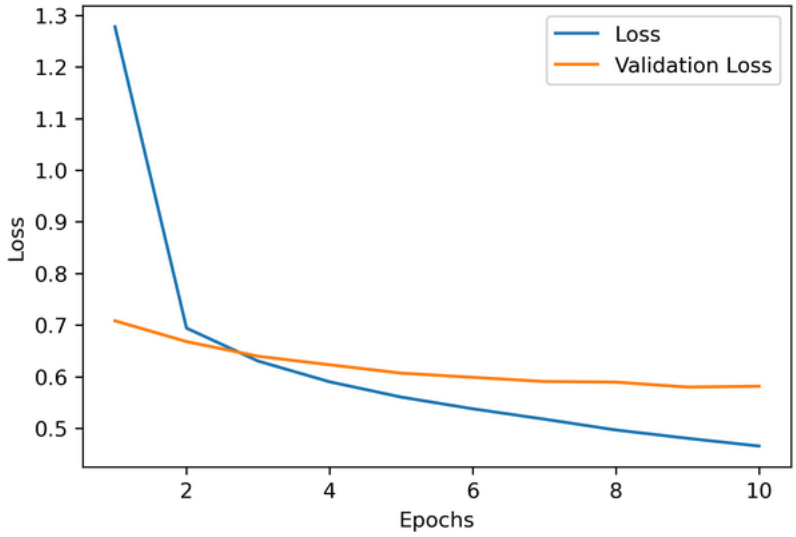

Figure 6: Loss Graph of T5 model for 10 Epochs

As we can in Figure 6, the model's loss was constantly decreasing which means the model was improving with each epoch. However, the validation loss was also decreasing slightly which means that the model will be able to generate summaries from unseen research papers. After training the model, ROUGE and BLEU scores of the generated summary were calculated which are the following:

Table 4: Evaluation Scores of T5

Metrics	T5 Model
ROUGE-1	80%
ROUGE-2	78%
ROUGE-L	80%
BLEU	49%

6.3 Evaluating T5 Model with Batch Size of 4 and Learning Rate of 1e-5

In this experiment, the T5 base model was applied to the preprocessed Scisumm dataset trained for 10 epochs. The batch size of 4 was chosen for this experiment and the learning rate of 1e-5 was selected because as we can see in experiment 2 the model was still slightly overfitting so in this experiment a small learning rate that 1e-5 was selected to measure the model's performance. From Figure 7, it can be seen that loss is decreasing with each epoch which means that the model is learning however, the validation loss is also decreasing which means that the model will be able to perform better when expose to unseen data and is the best trained T5 model.

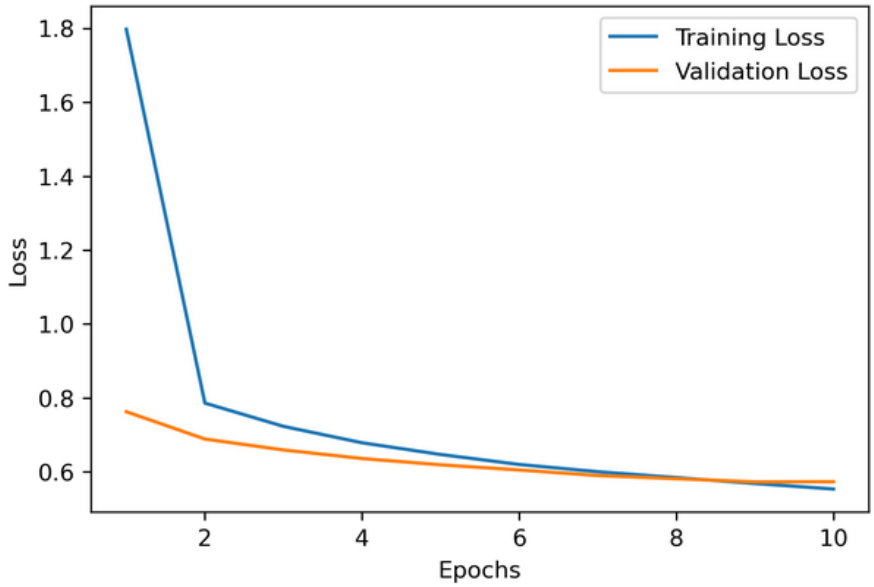

Figure 7: Loss Graph of T5 Model with a learning rate of 1e-5

The ROUGE and BLEU metrics were used to evaluate the performance of this model and from Table 5, we can see that this T5 model was performing better than the other models trained in previous experiments.

Table 5: Evaluation Scores of T5

Metrics	T5 Model
ROUGE-1	83%
ROUGE-2	82%
ROUGE-L	83%
BLEU	47%

6.4 Evaluating the BERT Model for Research Paper Summarization

In the fourth experiment, the BERT base model was trained with the preprocessed Scisumm dataset and trained with the same hyperparameters that were used to train the T5 model. The BERT model is the language model that is also based on the transformer architecture, the main reason to carry out this experiment was to check and compare the performance of the trained T5 model with the BERT model.

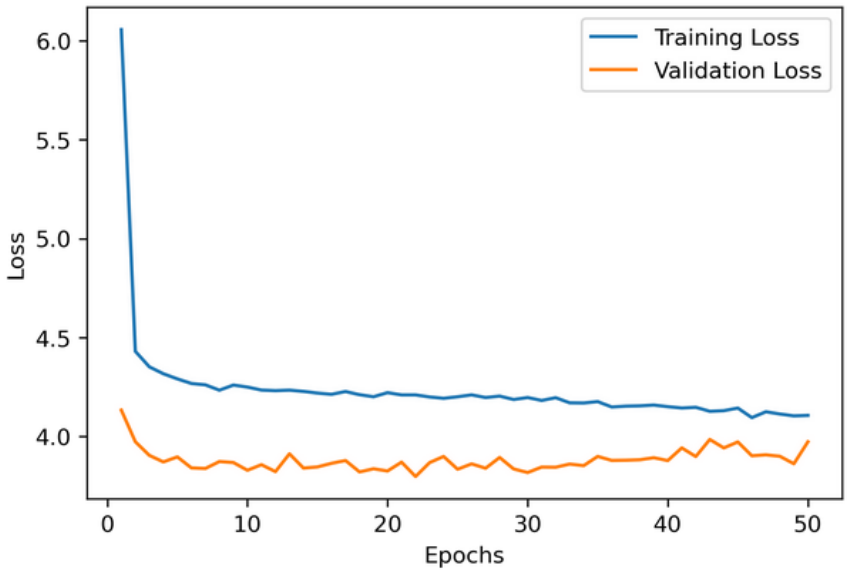

Figure 8: Loss Graph of the BERT Model for 50 Epochs

As we can in Figure 8, the loss of the BERT model slightly decreased with each epoch and there were some fluctuations with validation loss which was changing with each epoch. Moreover, it means that the BERT model was learning and improving while it was also performing well on unseen datasets but from the validation loss it was found that the data for validation was not enough for the model. The ROUGE and BLEU scores of the summaries generated by the trained BERT model were evaluated and compared with the T5 model which are available in Table 6. From Table 6, it can be seen that the trained T5 model was performing far better than the trained BERT model.

Table 6: Comparison of Evaluation Scores of T5 and BERT

Metrics	T5 Model	BERT Model
ROUGE-1	83%	54%
ROUGE-2	82%	37%
ROUGE-L	83%	43%
BLEU	47%	21%

6.5 Comparing T5 Model with BERT, GPT2 and BART

After training and evaluating the T5 model, the performance of the trained T5 model was compared with the other state-of-the-art models which were pre-trained for the text summarization tasks. The pre-trained base GPT2 and base BART models were downloaded from the Huggingface website. The ROUGE and BLEU scores were measured for all the models and as can be seen in Table 7, the trained T5 model with a learning rate of 1e-5 was performing better at generating the summaries of the research papers. The main reason the T5 model was able to surpass the other models like BERT, GPT2, and BART is due to its unified framework which means that T5 can use the past gained knowledge more effectively than any other model (Raffel *et al.*, 2020). Moreover, the T5 was trained on a large corpus of clean data while other models were trained on uncleaned and noisy data (Lewis *et al.*, 2019).

Table 7: Comparison of Evaluation Scores of T5 with GPT2, BERT, and BART

Evaluation Metrics	T5 Model	BERT Model	GPT-2 Model	BART Model
ROUGE-1	83%	54%	43%	67%
ROUGE-2	82%	37%	39%	56%
ROUGE-L	83%	43%	41%	62%
BLEU	47%	21%	20%	39%

6.6 Discussion

In this research project, the base T5 model was trained by tuning several hyperparameters like learning rate and batch size to find the best for the task of summarization of the research papers. Several experiments were carried out, in the first experiment the T5 model was trained for 50 epochs with a batch size of 8 to get the best result but after training it was found that the model was overfitting and was unable to perform well on unseen data as can be seen in Figure 5. To solve this issue, a new experiment was carried out in which the batch size of 4 was used and the model was trained for 10 epochs. Moreover, after training the model it was found that the T5 model was still slightly struggling when exposed to the unseen data but it was far better than the model trained in the first experiment as can be seen in Figure 6.

So, to further improve the model the learning was changed from 2e-5 to 1e-5 so that the model can take more time to learn from the training data as can be seen in Figure 7. Additionally, after evaluating the model it was found that this model was the most stable when expose to the new data and was generating good summaries of the research papers. As we can see in Table 1 in related studies most used metrics for evaluating the language models were ROUGE and BLEU so these two metrics were used in this research to measure the performance of the trained model.

After training the T5 model it was important to check how it is performing when compared to the other state-of-the-art models, so for this, the base BERT model was trained with the same hyperparameters to generate the summaries but the summaries generated by the BERT were not meaningful and did not get the good evaluation scores as can be seen in Table 6. So to futher evaluate the effectiveness of the trained T5 model it was compared with other pre trained models like GPT2 and BART. Moreover, the pre-trained models for text summarization like GPT2 and BART were downloaded from the Huggingface and the preprocessed data was passed through them. After evaluation it was found that the trained T5 model was able to outperform the other state-of-the-art models with the ROUGE-L Score of 83% and BLEU score of 47%. The reason the trained T5 model was able to perform better because is designed in such way that that it can transfer the knowledge it learned previously to new tasks. Furthermore, the T5 model was trained on large Corpus of clean text data while other models were trained on text which contains the noise and was not cleaned. The other reason is the T5 model contains both encoder and decoder layers as can be seen in Figure 3 while BERT and GPT2 only contain one layer. However, to further improve the T5 model a large dataset can be used and the long-form version of the T5 model can be used to generate more effective summaries. Some summaries generated by the best-trained T5 model with learning rate of 1e-5 are the following:

Table 8: Some Generated Summaries from the T5 Model

1	**Reference Summary:** parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques we present a stochastic parsing system consisting of a lexical-functional grammar (lfg) , a constraint-based parser and a stochastic disambiguation model . we report on the results of applying this system to parsing the upenn wall street journal (wsj) treebank . the model combines full and partial parsing techniques to reach full grammar coverage on unseen data . the treebank annotations are used to provide partially labeled data for discriminative statistical estimation using exponential models . disambiguation performance is evaluated by measuring matches of predicate-argument relations on two distinct test sets . on a gold standard of manually annotated f-structures for a subset of the wsj treebank , this evaluation reaches 79 % f-score . an evaluation on a gold standard of dependency relations for brown corpus data achieves 76 % f-score . we describe a discriminative lfg parsing model that is trained on standard (syntax only) tree bank annotations by treating each tree as a full lfg analysis with an observed c-structure and hidden f-structure . xle selects the most probable analysis from the potentially large candidate set by means of a stochastic disambiguation component based on a log-linear probability model that works on the packed representations .

	Generated Summary: parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques we present a stochastic parser based on an exponential model of linguistically fine-grained lfg , a constraint-based parse , and an exponential disambiguation model . the model combines full and partial parsers to reach full grammar coverage on unseen data . we use the wsj treebank annotations to provide partially labeled data for discriminative statistical estimation using exponential models . f-scores for both full and unlabeled parses , respectively
2	**Reference Summary:** cross-lingual word clusters for direct transfer of linguistic structure it has been established that incorporating word cluster features derived from large unlabeled corpora can significantly improve prediction of linguistic structure . while previous work has focused primarily on english , we extend these results to other languages along two dimensions . first , we show that these results hold true for a number of languages across families . second , and more interestingly , we provide an algorithm for inducing cross-lingual clusters and we show that features derived from these clusters significantly improve the accuracy of cross-lingual structure prediction . specifically , we show that by augmenting direct-transfer systems with cross-lingual cluster features , the relative error of delexicalized dependency parsers , trained on english treebanks and transferred to foreign languages , can be reduced by up to 13 % . when applying the same method to direct transfer of named-entity recognizers , we observe relative improvements of up to 26 % . **Generated Summary:** introducing cross-lingual word clusters for direct transfer of linguistic structure it has been established that incorporating word cluster features derived from large unlabeled corpora can significantly improve prediction of linguistic structure . while previous work has focused primarily on english , we extend these results to other languages , along two dimensions : first , they hold true for a number of languages . second , and more interestingly , a system for inducing crosslingual clusters and we show that by augmenting direct-transfer systems , the relative error of delexicalized dependency recognizers , can be reduced by up to 26 % .
3	**Reference Summary:** statistical machine translation by parsing in an ordinary syntactic parser , the input is a string , and the grammar ranges over strings . this paper explores generalizations of ordinary parsing algorithms that allow the input to consist of string tuples and/or the grammar to range over string tuples . such algorithms can infer the synchronous structures hidden in parallel texts . it turns out that these generalized parsers can do most of the work required to train and apply a syntax-aware statistical machine translation system . when a parser 's grammar can have fewer dimensions than the parser 's input , we call it a synchronizer . we formalize machine translation problem as synchronous parsing based on multi text grammars . **Generated Summary:** a statistical machine translation by parsing in an ordinary syntactic parser , the input is a string , and the grammar ranges over strings . this paper explores generalizations of ordinary parsers that allow the input to consist of string tuples and/or the grammar to range over string . such inference algorithms can perform various kinds of analysis on

	parallel texts , also known as multitexts . in this paper , we describe a synchronized mtg , which is based on the d-span vector . we use a multitree to infer a word-to-word

7 Conclusion and Future Work

In this research, the research papers were summarised using the Text-To-Text Transfer Transformer (T5) model, which was pre-trained on the large Corpus of text data to perform the NLP-related tasks. Furthermore, the base T5 model was trained on the Scisumm dataset which contains the 1009 research papers related to the sector of computer science with the reference summaries generated by the human experts. The main objective of this research was to check the performance of the T5 model when used specifically for the task of generating summaries of the research papers. To achieve this objective, the T5 model was tokenized using Auto tokenizer from the transformers library and then fine-tuned by changing several hyperparameters like batch size and learning rate and then evaluated using the metrics like BLEU and ROUGE. The trained T5 model got the ROUGE-1 score of 83%, ROUGE-2 score of 82%, ROUGE-L score of 83%, and BLEU score of 47% which means that the trained T5 model was able to generate good and meaningful summaires. Furthermore, the trained T5 model's performance was also compared with the other text summarization state-of-the-art models like BERT, GPT2, and BART and after the evaluation of the generated summaries, it was found that the trained T5 model was performing far better than the other models. However, the T5 model was limited by the issue of overfitting due to which hyperparameter tuning was done to get the best model.

In the future, the long-form version of the T5 model can be used because during this research when summaries were generated compiler was giving warnings due to the research papers variable lengths but was able to generate the summaries without any error. However, to solve this warning a longform version of the T5 was created so that research papers with variable lengths can be fed to the T5 model. Moreover, in this research, the papers related to the computer science field were used to further the scope of this study, papers from other sectors like medical or business can be used so that an application can be built in which the research papers of any sector can be directly fed to the T5 model to generate summaries quickly. Moreover, in the future the Scisumm dataset can be expanded with the help of the human experts to write the reference summaries and futher experiments can be conducted.

References

Luhn, H. P. (1958) "The automatic creation of literature abstracts," *IBM journal of research and development*, 2(2), pp. 159–165. doi: 10.1147/rd.22.0159.

Kupiec, J., Pedersen, J. and Chen, F. (1995) "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 68-73.

Neto, J. L., Freitas, A. A. and Kaestner, C. A. A. (2002) "Automatic text summarization using a machine learning approach," in *Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 205–215.

Steinberger, J. and Ježek, K. (2004) "Text summarization and singular value decomposition," in *Advances in Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 245–254.

Yeh, J.-Y. *et al.* (2005) "Text summarization using a trainable summarizer and latent semantic analysis," *Information processing & management*, 41(1), pp. 75–95. doi: 10.1016/j.ipm.2004.04.003.

Li, Y. and Yang, T. (2018) "Word embedding for understanding natural language: A survey," in *Studies in Big Data*. Cham: Springer International Publishing, pp. 83–104.

Zhang, Y. *et al.* (2019) "Abstract text summarization with a convolutional Seq2seq model," *Applied sciences (Basel, Switzerland)*, 9(8), p. 1665. doi: 10.3390/app9081665.

Adelia, R., Suyanto, S. and Wisesty, U. N. (2019) "Indonesian abstractive text summarization using bidirectional gated recurrent unit," *Procedia computer science*, 157, pp. 581–588. doi: 10.1016/j.procs.2019.09.017.

Rahman and Siddiqui (2019) "An optimized abstractive text summarization model using peephole convolutional LSTM," *Symmetry*, 11(10), p. 1290. doi: 10.3390/sym11101290.

Song, S., Huang, H. and Ruan, T. (2019) "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia tools and applications*, 78(1), pp. 857–875. doi: 10.1007/s11042-018-5749-3.

Malte, A. and Ratadiya, P. (2019) "Evolution of transfer learning in natural language processing," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1910.07370.

Moravvej, S. V., Mirzaei, A. and Safayani, M. (2021) "Biomedical text summarization using Conditional Generative Adversarial Network(CGAN)," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/2110.11870.

Tomer, M. and Kumar, M. (2020) "Improving Text Summarization using Ensembled Approach based on Fuzzy with LSTM," *Arabian journal for science and engineering*, 45(12), pp. 10743–10754. doi: 10.1007/s13369-020-04827-6.

Paulus, R., Xiong, C. and Socher, R. (2017) "A deep reinforced model for abstractive summarization," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1705.04304

Gupta, H. and Patel, M. (2021) "Method of text summarization using lsa and sentence based topic modelling with Bert," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, pp. 511–517.

Iwasaki, Y. *et al.* (2019) "Japanese abstractive text summarization using BERT," in *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, pp. 1–5.

Kieuvongngam, V., Tan, B. and Niu, Y. (2020) "Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/2006.01997.

Elsaid, A. *et al.* (2022) "A comprehensive review of Arabic text summarization," *IEEE access: practical innovations, open solutions*, 10, pp. 38012–38030. doi: 10.1109/access.2022.3163292.

Zhang, J. *et al.* (2020) "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," *arXiv [cs.CL]*. Edited by H. D. Iii and A. Singh. Available at: https://proceedings.mlr.press/v119/zhang20ae.html.

Ramina, M. *et al.* (2020) "Topic level summary generation using BERT induced Abstractive Summarization Model," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, pp. 747–752.

Bani-Almarjeh, M. and Kurdy, M.-B. (2023) "Arabic abstractive text summarization using RNN-based and transformer-based architectures," *Information processing & management*, 60(2), p. 103227. doi: 10.1016/j.ipm.2022.103227.

Miller, D. (2019) "Leveraging BERT for Extractive Text Summarization on Lectures," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1906.04165.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020) 'Exploring the limits of transfer learning with a unified text-to-text transformer', Journal of Machine Learning Research, ssrn.1533-7928

Ay, B. *et al.* (2023) "Turkish abstractive text document summarization using text to text transfer transformer," *Alexandria Engineering Journal*, 68, pp. 1–13. doi: 10.1016/j.aej.2023.01.008.

Garg, A. *et al.* (2021) "NEWS Article Summarization with Pretrained Transformer," in *Communications in Computer and Information Science*. Singapore: Springer Singapore, pp. 203–211.

Fendji, J. L. E. K. *et al.* (2021) "WATS-SMS: A T5-based French Wikipedia Abstractive Text Summarizer for SMS," *Future internet*, 13(9), p. 238. doi: 10.3390/fi13090238.

Chaurasia, S., Dasgupta, D. and Regunathan, R. (2023) "T5LSTM-RNN based text summarization model for behavioral biology literature," *Procedia computer science*, 218, pp. 585–593. doi: 10.1016/j.procs.2023.01.040.

Yasunaga, M. *et al.* (2017) "Graph-based Neural Multi-Document Summarization," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 452–462.

Yasunaga, M. *et al.* (2019) "ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1909.01716

Vaswani, A. *et al.* (2017) "Attention is all you need," *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1706.03762.

Lewis, M. *et al.* (2019) “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1910.13461.

Lin, C.-Y. (2004) “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81.

Kingma, D. P. and Ba, J. (2014) “Adam: A method for stochastic optimization,” *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1412.6980

Papineni, K. *et al.* (2001) “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL ’02*. Morristown, NJ, USA: Association for Computational Linguistics, pp. 311–318.

Grefenstette, G. (1999) “Tokenization,” in *Text, Speech and Language Technology*. Dordrecht: Springer Netherlands, pp. 117–133.

Dong, Y. (2018) “A survey on neural network-based summarization methods,” *arXiv [cs.CL]*. Available at: http://arxiv.org/abs/1804.04589.