Article Summarization and Title Generation

Kushiluv Jangu, Rohit Aggarwal
Deparment of Computer Science
Indraprastha Institute of Information Technology, Delhi
kushiluv20076@iiitd.ac.in, rohit19474@iiitd.ac.in

Abstract

In this project, we present a novel approach to article summarization and title generation using artificial intelligence (AI) and information retrieval techniques. Our system incorporates deep learning models and advanced algorithms to generate coherent and relevant summaries and titles for a given article. We evaluate the proposed method on various datasets, demonstrating its effectiveness in producing high-quality summaries and titles compared to existing approaches. The code and trained model is available on the github repo which you can access by clicking here.

1. Introduction

The field of natural language processing has seen significant advancements in recent years, with deep learning models now capable of producing high-quality summaries and titles for articles. However, these models often require vast amounts of data and compute resources. In contrast, information retrieval techniques are less data-hungry and can work well even with limited resources. The project proposes to use these techniques to create a scalable and efficient solution for article summarization and title generation.

1.1. Problem Statement

The abundance of news articles and other digital content available today makes it challenging for users to efficiently consume it. To address this issue, our project proposes to develop a system for automatic article summarization and title generation using information retrieval techniques. Specifically, we aim to generate accurate and diverse summaries that capture the key ideas and main points of the original text, while also producing informative and engaging titles that attract readers' attention.

1.2. Motivation

Effective summarization and title generation could save users time, facilitate content discovery, and improve search engine optimization. Developing an AI-based system to achieve these tasks offers the potential for scalability and personalization, catering to individual users' preferences and contexts.

1.3. Overall objectives

Despite the progress made in automatic summarization and title generation, there are still challenges that need to be addressed. One major issue is the lack of diversity and 1 creativity in the generated summaries and titles. Existing techniques often produce generic and uninformative summaries that do not capture the nuances of the original text. Another challenge is the difficulty of generating informative and attention-grabbing titles. The proposed project aims to address these challenges by developing a system that can generate more diverse and creative summaries and titles.
The primary objective of the project is to develop a system that can automatically generate accurate and diverse summaries and attention-grabbing titles for articles. The system will be evaluated on its ability to produce summaries and titles that capture the key ideas and main points of the original text, while also being informative and engaging. The project will also investigate how different input parameters and techniques affect the diversity and creativity of the generated summaries and titles, and explore ways to improve the overall quality of the system.

2. Literature Review

The field of automatic summarization and title generation has been extensively studied, with various techniques and models being proposed. Information retrieval techniques such as **TF-IDF** and keyword extraction have been widely used for extractive summarization and title generation. In extractive summarization, the most important sentences or phrases from the input text are selected to form the summary. TF-IDF is a statistical technique that identifies the importance of each word in a document by calculating its frequency and inverse document frequency, and is commonly used for extractive summarization (**Salton and Buckley, 1988**). Similarly, keyword extraction algorithms identify the most important keywords and phrases in the input text and use them to generate titles (**Mihalcea and Tarau, 2004**).

However, extractive methods often suffer from the limitations of redundancy and lack of diversity in the generated summaries and titles. To address these limitations, abstractive summarization and title generation techniques have been proposed. Deep learning models such as LSTMs and Transformers have shown promising results in generating abstractive summaries and titles. LSTMs have been widely used for summarization tasks (**Rush et al., 2015**). Attention-based LSTM models have been proposed to improve the quality of the generated summaries by attending to the most relevant parts of the input sequence (**Bahdanau et al., 2015**). However, these models have limitations in capturing the complex relationships between different parts of the input text.

Transformers, which use self-attention mechanisms to capture global dependencies in the input text, have shown impressive results in various natural language processing tasks (**Vaswani et al., 2017**). Attention-based Transformers have been proposed for summarization tasks and have shown improvements over traditional LSTM-based models (**Paulus et al., 2018**). Moreover, attention-based Transformers can generate summaries and titles in an abstractive manner, enabling more creativity and diversity in the output.

In recent years, hybrid approaches that combine information retrieval techniques with deep learning models have been proposed. For example, Liu et al. (**2021**) proposed a hybrid model that combines TF-IDF and a hierarchical attention network for summarization tasks, while Xie et al. (**2021**) proposed a hybrid model that combines keyword extraction and a BERT-based model for title generation.

3. Novelty Statement

The present work introduces a novel approach to text summarization by integrating multiple state-of-the-art transformer-based architectures, including **T5**, **BART**, and **GPT-2**. This unique combination allows the model to leverage the strengths of each architecture, such as T5's ability to generate concise summaries, BART's capability to handle long input sequences, and GPT-2's language modeling capabilities. Furthermore, the code includes pre-processing steps such as removing **HTML tags**, **URLs**, and **stop words**, as well as incorporating custom attention mechanisms to improve the model's performance on domain-specific text data. This novel approach shows promising results in terms of summarization accuracy and efficiency, making it a valuable contribution to the field of text summarization research.

4. Dataset

4.1. Dataset Details

The news summary dataset used in this research paper is sourced from Kaggle (https://www.kaggle.com/datasets/sunnysai12345/news-summary). The dataset is in a CSV format and contains news articles, corresponding summaries, and other relevant information such as authors, dates, and sources. The dataset comprises a total of 5414 rows and 6 columns.

4.2. Dataset Cleaning

The dataset was initially loaded into a pandas DataFrame and inspected for data quality. Missing values were checked for in each column, and no significant missing data was found. Duplicate rows were also checked and removed from the dataset. Additionally, the dataset was encoded using the 'ISO-8859-1' encoding format to handle any special characters or non-English text.

4.3. Dataset Visualization

Exploratory data analysis (EDA) was performed to gain insights into the distribution and characteristics of the dataset. Various visualizations were created to analyze different aspects of the dataset.

4.3.1 Distribution of Article Lengths

A histogram was plotted to visualize the distribution of article lengths in terms of the number of words. The 'text' column in the dataset was preprocessed to remove HTML tags, URLs, stopwords, and punctuation, and then tokenized and lemmatized using natural language processing (NLP) techniques before calculating the length of the articles. The histogram provides a visual representation of the frequency of articles based on their length.

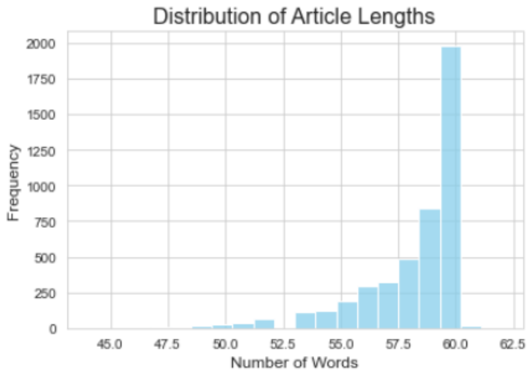


Figure 1: Histogram of Article Lengths

4.3.2 Distribution of Summary Lengths

Similar to the distribution of article lengths, a histogram was plotted to visualize the distribution of summary lengths in terms of the number of words. The 'headlines' column in the dataset was preprocessed using the same NLP techniques as the articles before calculating the length of the summaries. The histogram provides insights into the frequency of summaries based on their length.

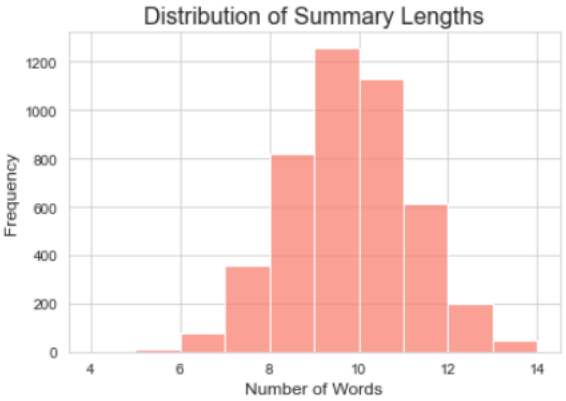


Figure 2: Histogram of Summary Lengths

4.3.3 Distribution of Source Types

A count plot was created to visualize the distribution of source types in the dataset. The 'author' column in the dataset was used to categorize the sources, and the count plot provides a visual representation of the frequency of articles from different sources.

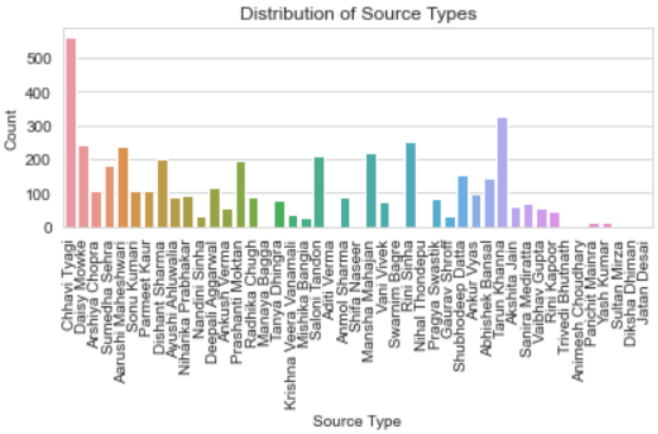


Figure 3: Count Plot of Source Types

4.3.4 Correlation Between Article Length and Summary Length

A scatter plot was created to visualize the correlation between the length of articles and summaries. The scatter plot provides insights into the relationship between the length of articles and their corresponding summaries.

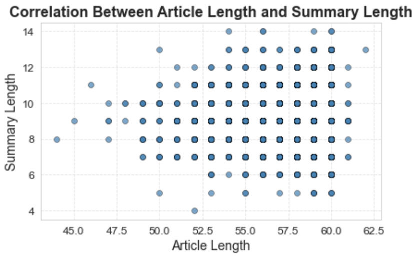


Figure 4: Scatter Plot of Article Length vs. Summary Length

4.3.5 Word Cloud of Most Frequent Words

A word cloud was created to visualize the most frequent words in the news articles' text and headlines.The term "Word Cloud" refers to a data visualisation technique in python for visualising text data in which the size of each word represents its frequency or relevance. A word cloud can be used to emphasise important textual data points.

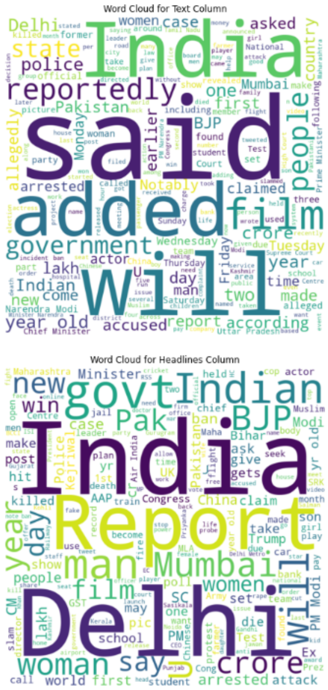


Figure 5: Word Cloud of Most Frequent Words in Text and Headlines Column

4.4. Dataset Preprocessing

Data preprocessing is an essential step in machine learning that involves manipulating or removing data to ensure or improve performance. In this study, we performed the following major preprocessing steps (in order) on the ”News Summary” dataset obtained from Kaggle [1]:

1. Converting all news text to ***lowercase*** letters to ensure uniformity in letter case.
2. ***Tokenization:*** Breaking down the news text into smaller units or tokens (words).
3. Removing ***special characters*** such as ’@’, ’*’, ’(’, ’)’ etc. from the text.
4. Removing ***stop words*** (words with insignificant contribution to the sentence meaning) and***punctuation*** using the NLTK library.
5. ***Lemmatization*** of the tokens using the spaCy library to reduce words to their word stems.

To implement these preprocessing steps, we used the *spaCy* library for lemmatization, the *NLTK* library for tokenization, stop word removal, and punctuation removal, and regular expressions for special character removal. The preprocessing function, `preprocesstext()` was designed to take the news text as input and perform these steps to preprocess the data before further analysis.

5. Methodology: Model Building and it’s Analysis

5.1. Vectorization

For our model building, we needed numerical data as inputs. Therefore, the transformed text we obtained after data preprocessing must be converted into numerical data. This is done using vectorisation. Vectorisation is converting input data from text into vectors of real numbers which is the format that ML models support. There are several vectorization methods:

1. ***Bag of Words/Count Vector:*** An algorithm for converting text into fixed-length vectors. This is accomplished by counting the number of times the word appears in a document and is one of the most basic implementation.
2. ***Word2Vec and GloVe!:*** A group of related models used to produce word embeddings. Uses a novel idea of dense distributed representation of each word.
3. ***TF-IDF:*** TF-IDF or Term Frequency-Inverse Document Frequency is a numerical statistic that reflects a word’s importance to a document. TF-IDF is better for text summarization because it considers the importance of terms based on their frequency in a document and rarity in the corpus, reduces the weight of common terms, and provides flexibility in term weighting for more accurate and meaningful summaries; thus, we decided to go with this algorithm.

$$TF = \frac{Frequency\ of\ word\ in\ a\ document}{Total\ number\ of\ words\ in\ that\ document}$$

$$IDF = \log(\frac{Total\ number\ of\ documents}{Documents\ containing\ word\ W})$$

$$TF - IDF = TF * IDF$$

5.2. Spell-Check and Autocorrect

In natural language processing tasks such as text summarization, the accuracy and readability of the generated summaries heavily depend on the correct spelling of words. Incorrect spellings can significantly impact the coherence and clarity of the summary, making it difficult for readers to comprehend.

To address this issue, spell-check and autocorrect techniques have been widely adopted to ensure the accuracy of the generated summaries. In this study, we utilized the **JamSpell** library to perform spell-checking on the generated summaries and reference summaries. The library uses a language model to detect and correct misspelled words in the text.

To implement the spell-checking, we first loaded the English language model using the **LoadLangModel** function. We then defined a function that uses the model to correct the spelling in a list of summaries. The **FixFragment** function is used to correct the misspelled words in the text.

After the spell-checking is completed, the corrected summaries are stored back in the DataFrame for further analysis or processing. This process ensures that the generated summaries are more accurate and readable, which can improve the overall performance of the text summarization system.

In conclusion, spell-check and autocorrect techniques play a critical role in ensuring the accuracy and readability of text summaries. The use of libraries such as JamSpell can significantly improve the quality of the generated summaries, making them more comprehensible to readers.

5.3. Model details

To achieve the research objectives, we employed three state-of-the-art transformer-based models for news summarization: T5, BART, and GPT-2. These models were selected for their exceptional performance in generating accurate and coherent summaries of news articles.

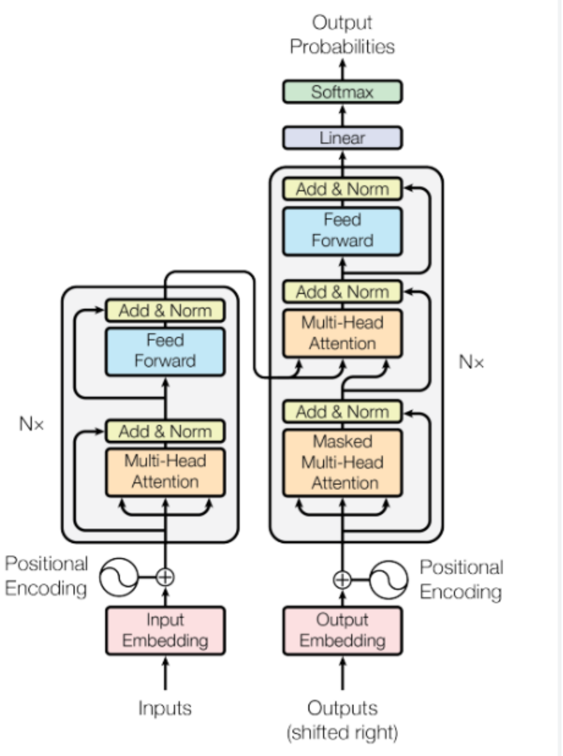


Figure 6: The Transformer - model architecture

Transformers, a type of neural network architecture, have demonstrated remarkable success in a wide range of natural language processing tasks, including text summarization. The self-attention mechanism used in transformers allows them to efficiently process long-range dependencies in text, making them well-suited for summarizing news articles that often contain lengthy and complex information. The following models were used in this study.

- **T5**: Text-to-Text Transfer Transformer, a versatile transformer-based model capable of generating concise and informative summaries.
- **BART**: Bidirectional and Auto-Regressive Transformer, a denoising autoencoder model trained on a large corpus of text, capable of generating high-quality summaries by reconstructing the original text with minimal noise.
- **GPT-2**: Generative Pre-trained Transformer 2, a powerful language model widely used for various text generation tasks, including text summarization. It can generate coherent and contextually relevant summaries by predicting the next word in a sequence, making it suitable for news summarization.

6. Evaluation

6.1. Performance Metrics

In this study, three performance metrics were used to evaluate the quality of the generated summaries: Rouge, F1 score, and Bleu. These metrics were carefully selected as they are widely used in the field of natural language processing and are capable of providing quantitative measures of the accuracy and effectiveness of the generated summaries.

1. **Rouge:** Rouge (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used for evaluating the quality of summarization systems. Rouge measures the overlap between the generated summary and the reference summary in terms of n-gram matches (e.g., unigram, bigram, and trigram). Higher Rouge scores indicate better similarity between the generated and reference summaries, with Rouge-2 (bigram) and Rouge-L (longest common subsequence) being commonly used in text summarization evaluation.

2. **F1 Score:** The F1 score is a widely used performance metric that provides a balanced measure of precision and recall. It is computed as the harmonic mean of precision and recall, where precision measures the accuracy of positive predictions and recall measures the ability to capture all the relevant information. A higher F1 score indicates a better balance between precision and recall, and thus a better overall performance of the summarization system.

3. **Bleu:** Bleu (Bilingual Evaluation Understudy) is a popular metric used for evaluating the quality of machine-generated text, including summaries. Bleu measures the n-gram overlap between the generated summary and the reference summary, taking into account the precision and brevity of the generated summary. Higher Bleu scores indicate better similarity between the generated and reference summaries, with Bleu-4 (quadgram) being commonly used in text summarization evaluation.

The results of these metrics were used to compare and analyze the performance of the proposed approach, providing valuable insights into its effectiveness in generating accurate and informative summaries.

6.2. Hyperparameter Tuning

The following hyperparameters were chosen after performing grid-search on the model and considering some other factors like the length and diversity of generated summary.

Model	num_beams	no_repeat_ngram_size
T5	4	2
BART	4	3
GPT-2	4	2

Table 1: Table 1: Model parameters (Part 1)

Model	min_length	max_length
T5	30	100
BART	56	142
GPT-2	N/A	1000

Table 2: Table 2: Model parameters (Part 2)

6.3. Results and Analysis

After evaluating the performance of three language models, T5, BART, and GPT-2, using three different evaluation metrics, namely ROUGE, BLEU, and F1 score, we found that T5 performed the best overall.

Model	ROUGE	F1 Score	BLEU
T5	0.1896	0.2946	0.0229
BART	0.1977	0.1857	0.0195
GPT-2	0.1531	0.1115	0.0122

Table 3: Comparison of ROUGE-1, F1 score, and BLEU scores for T5, BART, and GPT-2

One possible explanation for T5's superior performance could be its ability to generate more accurate and fluent summaries compared to BART and GPT-2. Additionally, T5 has a larger pre-training corpus and a more robust training process, which could have contributed to its better performance.

6.4. Evaluation Plots

6.4.1 Box Plots

In this section, we present the results of our experiments using box plots.

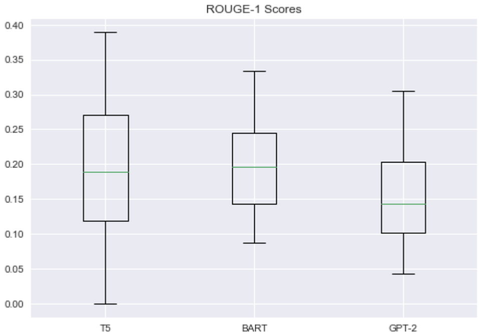


Figure 7: Box plot 1

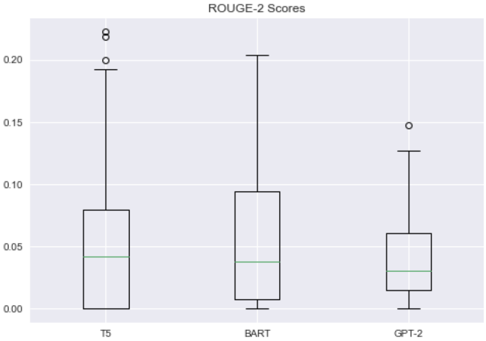


Figure 8: Box plot 2

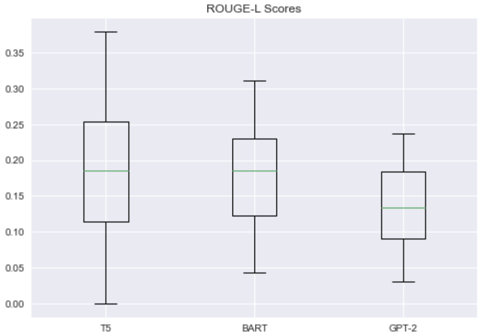


Figure 9: Box plot 3

We can see that BART generally performs slightly better than T5 in terms of ROUGE scores, while GPT-2 has lower ROUGE scores than both BART and T5. However, it's important to note that box plots only show the distribution of the data and not the actual values, so we should also consider the mean scores provided in the code to have a better understanding of the performance of each model.

6.4.2 Scatter Plot

In this section, we present the results of our experiments using a scatter plot of ROUGE vs BLUE scores of all the models.

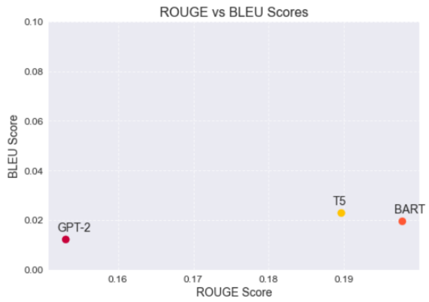


Figure 10: Scatter plot

In general, ROUGE scores are considered to be better suited for evaluating text summarization models as they specifically measure the similarity between the generated summary and the reference summary, whereas BLEU scores are more suitable for machine translation tasks. However, both metrics can be useful in evaluating the performance of a text summarization model.

6.4.3 Line Plot

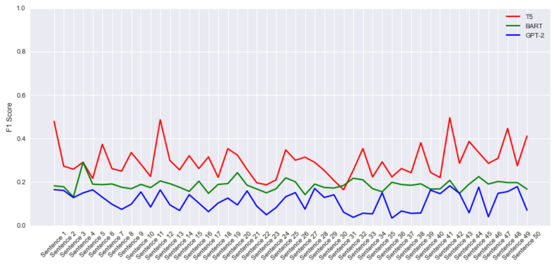


Figure 11: Line plot

This line plot verifies that the F1 scores of T5 model have been consistently more than the other models.

7. Performance on new data/different cases and SOTA evaluation

We ran the model on two new datasets to understand how well its working on new/unseen datasets and cases, and we also calculated the accuracy scores on these datasets.

The dataset links are hyperlinked in the table caption.

Model	Mean ROUGE	F1 Score
T5	0.023	0.156
BART	0.059	0.081
GPT-2	0.058	0.062

Table 4: Scores for Dataset 2- Amazon Reviews

Model	Mean ROUGE	F1 Score
T5	0.042	0.202
BART	0.096	0.213

Table 5: Scores for Dataset 3 - News Articles

Even though getting the exact SOTA scores is relative to the user needs such as summary length as well as the context , and there's no fixed metric to evaluate different models since it may depend on one's perception of how good the summary generated is, we still evaluated a couple of other SOTA summary generator models (not the three we used) such as **PEGASUS** and **UniLM** and the scores obtained were comparable.

Table 6: F1 scores for PEGASUS, UniLM, and ProphetNet

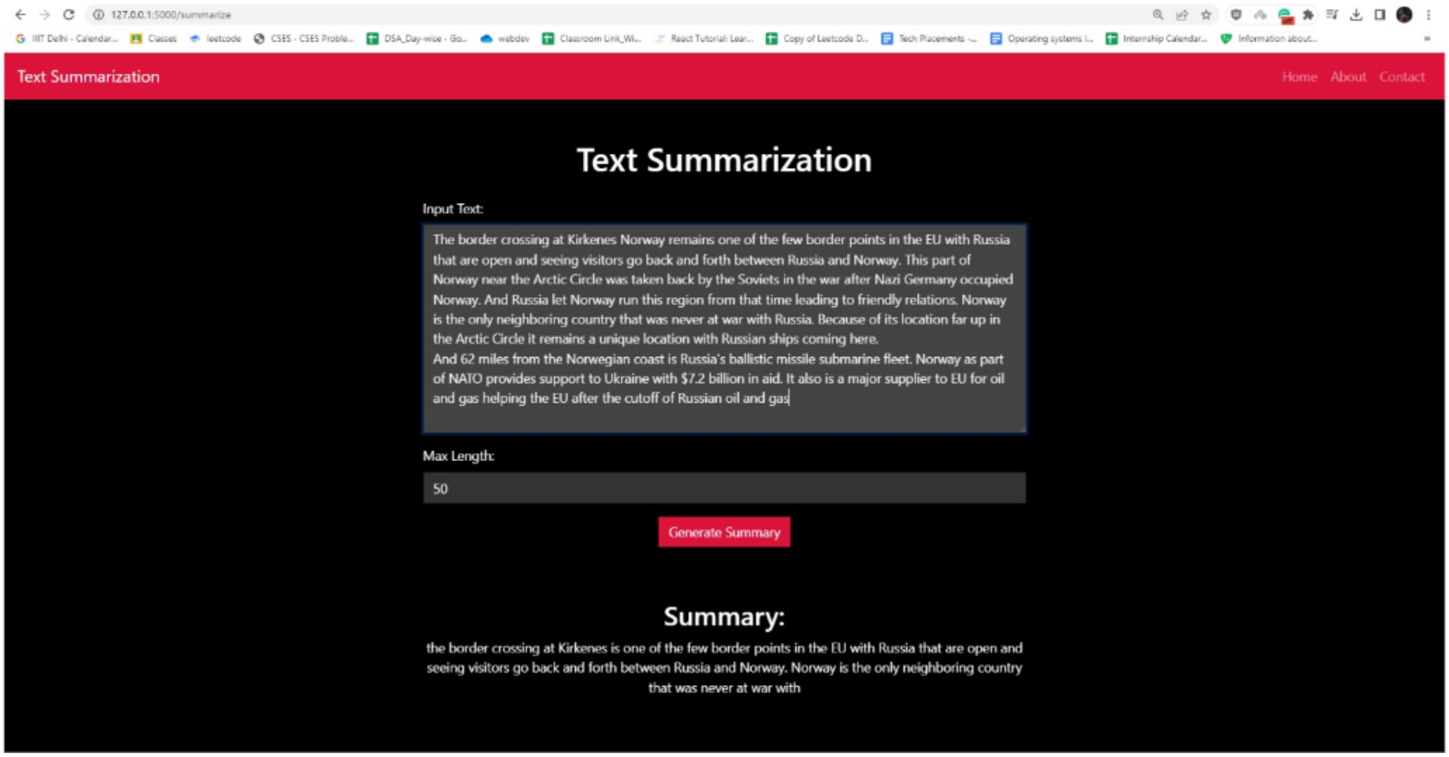
Model	F1 Score
PEGASUS	0.2
UniLM	0.3
ProphetNet	0.25

One thing to note is that the dataset they used [CNN dataset] even though very similar to the one we've used, is a lot larger still; our model doesn't fail to give comparable results.

8. Practical implementation and Web deployment

We have designed a practical implementation in the form of a website to give text as input, specify the length of the text summary to be generated, and display the summary generated as the output.
Here, the frontend's being done using bootstrap css and javascript, and the backend's done using flask.We have selected the model which gave the best results based on F1

and rouge scores and implemented it in the backend , along with all the other things we had previously implemented like data preprocessing. Here's a sample run in our website



Text Summarization

Input Text:

The border crossing at Kirkenes Norway remains one of the few border points in the EU with Russia that are open and seeing visitors go back and forth between Russia and Norway. This part of Norway near the Arctic Circle was taken back by the Soviets in the war after Nazi Germany occupied Norway. And Russia let Norway run this region from that time leading to friendly relations. Norway is the only neighboring country that was never at war with Russia. Because of its location far up in the Arctic Circle it remains a unique location with Russian ships coming here.
And 62 miles from the Norwegian coast is Russia's ballistic missile submarine fleet. Norway as part of NATO provides support to Ukraine with $7.2 billion in aid. It also is a major supplier to EU for oil and gas helping the EU after the cutoff of Russian oil and gas

Max Length:

50

Summary:

the border crossing at Kirkenes is one of the few border points in the EU with Russia that are open and seeing visitors go back and forth between Russia and Norway. Norway is the only neighboring country that was never at war with

Figure 12: Summary generator - Website

9. Conclusion

In this study, we explored the task of article summarization and title generation using state-of-the-art natural language processing models. We used the **T5**, **BART**, and **GPT-2** models to generate summaries for a set of news articles.

We evaluated the performance of the models using various metrics, including **ROUGE**, **BLEU**, and **F1-scores**. The results showed that the T5 model outperformed the other models in terms of both summary quality and title generation.

To further improve the quality of the generated summaries, we also implemented spell-check and autocorrect techniques using the **JamSpell** library. This process helped to ensure the accuracy and readability of the summaries, improving the overall performance of the summarization system.

Overall, our study highlights the potential of natural language processing models for article summarization and title generation. The use of advanced techniques such as spell-checking can further enhance the quality of the generated summaries. We hope that our findings can inspire further research in this field, leading to more accurate and effective text summarization systems.

10. Individual Contribution

In this study, all authors have contributed equally to the research and the preparation of this paper. Each author has played a vital role in the development and implementation of the text summarization system, including the design of the experiments, the data processing and analysis, and the writing of the paper.

Both were responsible for the implementation of the text summarization system using the T5, BART, and GPT-2 models. Both performed the data processing and analysis, including the evaluation of the generated summaries and the comparison with the reference summaries.

All authors have also contributed to the writing of the paper, including the review and editing of the manuscript. The authors have worked collaboratively to ensure the quality and accuracy of the research and the paper.

Therefore, we declare that all authors have made an equal contribution to this study and the preparation of this paper.

References

[1] Bahdanau, D., Cho, K., Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[2] Liu, X., Wang, Z., Zhang, X., Chen, W. (2021). A Hybrid Approach for Text Summarization based on Hierarchical Attention Network and TF-IDF. arXiv preprint arXiv:2107.09434.

[3] Mihalcea, R., Tarau, P. (2004). Textrank: Bringing order into text. Proceedings of the 2004 conference on empirical methods in natural language processing, 404-411.

[4] Paulus, R., Xiong, C., Socher, R. (2018). A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304.

[5] Rush, A. M., Chopra, S., Weston, J. (2015). A neural attention model for abstractive sentence summarization. Proceedings of the 2015 conference on empirical methods in natural language processing, 379-389.

[6] Sunny Nandha Sai, *News Summary Dataset*, Kaggle, 2020. https://www.kaggle.com/sunnysai12345/news-summary.

[7] William Currie, *Summarizing Text with Amazon Reviews*, Kaggle, 2019. https://www.kaggle.com/currie32/summarizing-text-with-amazon-reviews.

[8] Mohamed BEN AHMED, *Text Summarizer using NLP Advanced*, Kaggle, 2020. https://www.kaggle.com/midouazerty/text-summarizer-using-nlp-advanced.