Abstractive Text Summarization using Transformer Architecture

Shubham Dhapola
Computer Science and Engineering
Graphic Era Hill University
Dehradun, India
dhapolashubham@gmail.com

Siddhant Goel
Computer Science and Engineering
Graphic Era Hill University
Dehradun, India
goelsiddhant843@gmail.com

Daksh Rawat
Computer Science and Engineering
Graphic Era Hill University
Dehradun, India
iamdakshrawat@gmail.com

Satvik Vats, SMIEEE
Computer Science and Engineering,
Graphic Era Hill University;
Adjunct professor, Graphic Era Deemed to be University,
Dehradun, 248002, India.
svats@gehu.ac.in

Vikrant Sharma, SMIEEE
Computer Science and Engineering,
Graphic Era Hill University;
Adjunct professor, Graphic Era Deemed to be University,
Dehradun, 248002, India.
vsharma@gehu.ac.in

***Abstract*— The internet has large amounts of textual data which has become a double-edged sword. The information is always available but due to its high volume it is hard to find the things which we are looking for. In the past, subject experts were used to create comprehensions which was very time consuming and not financially viable. These comprehensions were made by reading lengthy texts, but often gave redundant results when working on similar articles. To address this problem, Automatic Text Summarization can be used. It helps in generating concise summaries using information retrieval by capturing the document's essence which makes it easier for users to find the key points from the original text. Imagine quickly grasping an article's core message before deciding on a deeper dive. This efficiency is invaluable in today's information-driven world. This paper proposes an advanced "abstractive" text summarization system that delves deeper than simply extracting key sentences. By leveraging Natural Language Processing techniques, the system analyses the text's structure, identifies salient points, and grasps their relationships. By comprehending the underlying context and logic, the system generates summaries using novel phrasings that accurately reflect the content, paving the way for a revolution in information navigation.**

Keywords—Summarization, Natural Language Processing, Transformers, Deep Learning

I. Introduction

Every day, we encounter vast amounts of information. The maximum part of this information is in the form of text, written articles, blogs, newspapers, etc. To understand the content of these articles, we need to read each article thoroughly. Longer articles can be time consuming to comprehend. Articles which are based on similar topics require more efforts to extract the non-redundant information due to the information overlap. This paper discusses an approach of abstractive text summarization which reads an entire document and generates the summary which contains the essence of the original document. The main goal of text summarization is to develop a system that can process the article's data using different Natural Language Processing techniques. This method helps in reducing the time taken by an individual to read and understand the lengthy articles.

Due to the internet explosion, a large amount of data is at our fingertips, but is has also created the challenge information overload. There is much more data present than we can process, but finding what we need is a struggle. There are many processes and methods that are used to extract the relevant information from the text like search engines, recommendation systems, and question-answering systems. Additionally, there are different techniques like text summarization and information extraction that simplify the complex information by extracting key points and considering lengthy text, making it simpler and easier to understand.

Automatic text summarization faces several hurdles, including pinpointing the main topic, analyzing the content, crafting the summary for the text, and judging the quality of the model. Currently, most of the systems rely on the "extractive" approach of text summarization, which only finds out the key sentences from the text and gives the summary by combining them. However, there is a need for "abstractive" text summarization models. This model will understand the actual meaning of the input text and then summarize it by making meaningful sentences. This technology would be a game-changer for navigating the ever-growing sea of online information. This paper aims to develop a system that can automatically generate the summary for the given text with better accuracy by understanding the core meaning of the text and not by just copying existing sentences.

II. Literature Survey

K. D. Garg et.al [1] They describe text summarization as shortening the text document while preserving its context and general idea. Every summary should have the main idea of the main text document. They have talked about using several NLP techniques to extract text from the documents.

979-8-3503-8459-8/24/$31.00 ©2024 IEEE
DOI: 10.1109/AIC.2024.3

They have used an unsupervised machine learning technique to develop a Punjabi text summarizer which is an extractive summarizer. They suggested various processes like tokenizing the text, extracting stop words from it, finding similarity matrix, ranking using similarity matrix, and creating a summary.

K. Mona Teja et.al [2] Blind people read text using different ways and one of them is Braille script, which is very ineffective since it is very time consuming and requires a lot of skill. The authors have proposed a new solution to help visually impaired people. Their research tries to summarize news articles to decrease the reading time. They compared various methods like Text Ranking Algorithm, Luhn's Algorithm and Latent Semantic Analysis Algorithm in their paper.

Tacha Jo [3] A modified KNN approach has been proposed in which words are treated as features. Their research approaches the text summarization task as a classification problem. Their approach involves dividing the original document into small chunks of paragraphs and sentences and every chunk is then classified as summary and non-summary. The chunks which are classified as part of the summary are then selected to form the final summary. The modified KNN shows a better performance and more compact representation of data.

N. Moratanch et. al [4], Their research compares various abstractive text summarization techniques. There are two main approaches, the first is Semantic based abstractive text summarization and the other is Structure based abstractive text summarization. The author has also discussed various techniques and challenges they came across while implementing abstractive text summarization approaches.

Pankaj Gupta et. al [5], analyzed various techniques used for text summarization and sentiment analysis. Their approach involves extracting the emotions of the text. The author has used two machine learning algorithms which are Naïve Bayes Classifier and Support Vector Machines. These algorithms are used for sentiment analysis of the text. Text Summarization leverages Natural Language Processing and semantic characteristics of the sentences to find the importance of words and sentences to include them in the final summary. This paper presents a survey of research done in the field of text summarization and sentiment analysis by evaluating pros and cons of different strategies and techniques.

Dharmendra Hingu et. al [6], The author has discussed extractive text summarization in their paper. They used Wikipedia articles as an input in their system to identify text scoring. In the initial step, sentences are tokenized by pattern matching with regular expressions. Orthodox methods are used to score the sentences, which helps in the classification task of whether they should be present in the final result or not. Scoring is helpful for identifying the words that can be used in sentences for generating accurate summaries. It results that the scoring sentences that are based on the citation gives the better output.

C. Prakash et. al [7] Because of the digital revolution, huge amount of data is available online, but is not very accurate. The search engines provide huge amounts of data which can't be processed by humans. To do that we need to get the abstract of the data without having to read the whole document. 'SAAR' a human aided text summarizer is proposed which summarizes a single document. If the generated summary is liked by the user, they can finalize it or else the model will again generate a new summary. The performance was tested using different metrics precision, recall and F1-score.

Kavita Ganesan et. al [8], Introduced Opinosis, a graph-based framework for abstractive text summarization. This framework is aimed at processing data like documents, movies, reviews. The approach involves constructing an Opinosis graph from the text where nodes represent words. This framework utilizes different graph properties, such as collapsible structures, redundancy capture and gapped subsequence capture helps in generating abstractive summaries. In the graph generated, those paths are selected which are valid and marked with high redundancy score. The selected paths are then ordered in the descending value of their redundancy scores. Jaccard index is used to remove the duplicate paths for comparing it with the human generated summaries.

Chin-Yew Lin et. al [9], This research paper first introduced ROUGE (Recall Oriented Understudy for Gisting Evaluation). It is a set of metrics used to evaluate the quality of summaries by comparing them to reference summaries. It is widely used in natural language processing tasks. Various types of ROUGE scores are ROUGE-L, ROUGE-N, ROUGE-W and ROUGE-S. ROUGE methods are effective for evaluating both single and multiple document summaries.

III. METHODOLOGY

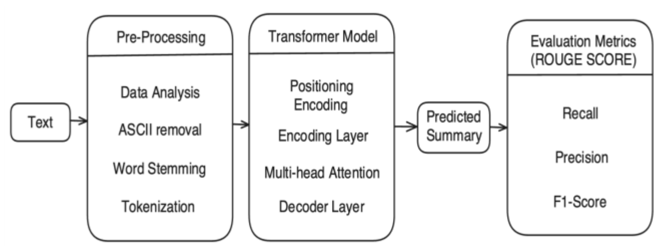

Figure 1: Methodology and its workflow

Figure 1 shows the overall workflow of the project. Initially, the text is pre-processed and then given to the transformer model for generating the summarized text. Then we evaluate the output using ROUGE scores.

A. Pre-Processing

Data preprocessing is used for cleaning and transforming raw data into the format which is required for analyzing. Data preprocessing is used to reduce noise, for enhancing the quality of data.

Data preprocessing involves handling missing values, handling categorical data, feature engineering and splitting the dataset [10]. Tokenization is an important task in the model. It involves converting words into tokens. The maximum and the minimum length of the summary is determined at this stage.

B. The Transformer Architecture

Transformer is an important deep learning architecture which was introduced by Google researchers in "Attention is All You Need" paper by Vaswani et al. The main goal of this architecture was to help in natural language processing tasks but has been applied to other domains also by creating suitable embeddings.

The Transformer architecture is a bit advanced then RNNs and LSTMs. It relies on a special kind of attention mechanism called self-attention. Self-attention helps in finding global dependencies between input and output sequences. This makes parallelization possible and efficient, which is not possible in RNNs and LSTMs.

The transformer architecture is utilized in various techniques of Natural Language Processing. It solves the tasks sequentially, and, at the same time, it successfully deals with the long-term dependencies. It is important to note that prior to the multi-head attention operation, there are several attentions layers existing in the transformer's architecture. These apply in highlighting some features to recognize, the pattern of the words in each input dataset are acknowledged by the method of the positional encoding.

It also reinforces in minimizing the time taken to process different sets of data at the same time. The adjustment of the transformer network therefore departs from the RNN process for the filtration of dataset [11]. Instead, it employs the layers that correspond to it by connecting the multi-head attention layer and the feed-forward network layers. Speaking of several types of attention, self-attention or intra-attention is the process of comparing different positions within a single pattern in order to compute a sequential design. In the transformer network, both the encoder & decoder models gain the ability of the attention mechanism with the help of 'multi-head attention' layer. Transformer models utilize encoder-decoder roles for the text summing up process.

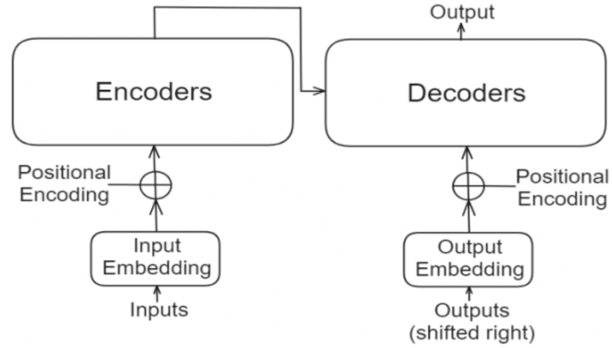

Figure 2: Basic Transformer Architecture

The above diagram shows the major components of the Transformer architecture that are encoder and decoder. It is not necessary to use both the components in the Transformer architecture.

1.Encoder: It is concerned with the manipulation of input sequences and the creation of a useful representation of it. The encoder primarily consists of several layers of feed-forward neural networks, and self-attention mechanisms. Each layer in the encoder is an independent layer and information flows between these layers parallelly. The output of the encoder is a collection of contextualized representations for each token in the input sequence, which captures both the local and global dependencies.

2.Decoder: It is responsible for generating output sequence based on the input received from the encoder. The decoder consists of multiple independent layers which are self-attention and feed-forward neural network layers. During the decoding process the encoder's output and the previously generated tokens are used for generating new tokens.

3.Attention: It is the foundation of the Transformer architecture and is used to capture dependencies between different parts of input and output sequences. It computes weighted sum of input representations, where the weights are based on the relevance of each input token to the current token being processed. Attention mechanisms help transformers to capture long-range dependencies and improve performance on various sequence to sequence tasks.

The encoder analyses the text, converting words into numerical representations and understanding relationships between them. This creates a compressed understanding of the document. The decoder then uses this information and attention to focus on key parts, building a concise summary word by word. This method allows for faster processing and superior grasp of complex documents compared to traditional techniques.

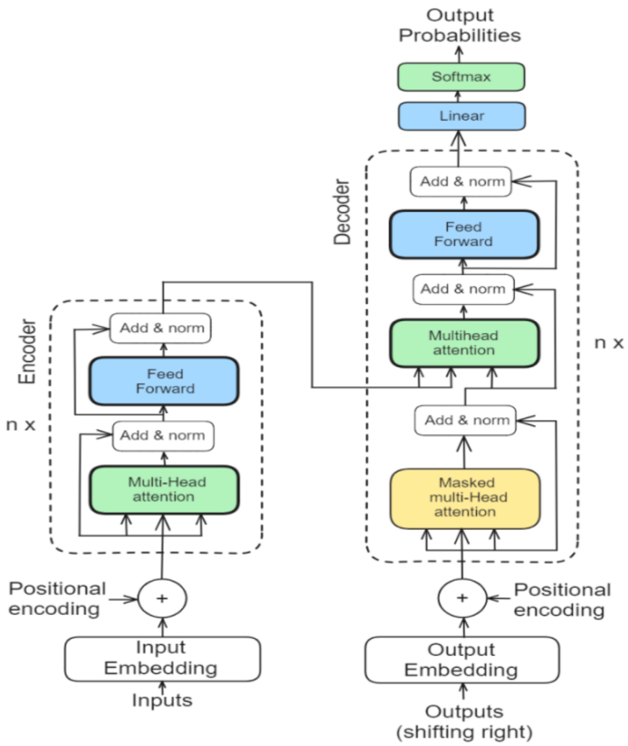

Figure 3: Transformer Architecture

The above diagram shows different layers of encoder and decoder in the Transformer architecture. The encoder uses self-attention to process input sequences and generate embeddings, while the decoder uses self-attention and cross-attention to produce output sequences, enabling efficient parallel processing and capturing long-range dependencies.

Natural language processing technique is used in the process of Abstractive summarization for the processing of input data. A new summarized document is generated from the input document by extracting useful and important information using this architecture.

C. *Evaluation Metrics*

ROUGE is a set of metrics used to evaluate the quality of summaries by comparing them to reference summaries. It compares the extent of similarity between the generated summary or translation and the reference summaries or translations.

1.ROUGE-N (unigram, bigram, n-gram): It measures the similarity of n-grams in the system generated summary and the reference summary.

Precision:

$$Precision = \frac{NgO}{NgGS} \quad (1)$$

Where, NgO = Number of Overlapping n grams, NgGS = Number of n grams in the generated Summary

Recall:

$$Recall = \frac{NgO}{NgRS} \quad (2)$$

Where, NgO = Number of overlapping n grams, NgRS = Number of n grams in the reference summary

F1-Score :

$$= \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

2.ROUGE-L: Measures the longest common subsequence (LCS) between the system and reference summaries.

Precision:

$$Precision = \frac{LenLCS}{NwGS} \quad (4)$$

Where, LenLCS = Length of longest common subsequence, NwGS = Number of words in the generated summary

Recall:

$$Recall = \frac{LenLCS}{NwRS} \quad (5)$$

Where, LenLCS = Length of longest common subsequence, NwRS = Number of words in the reference summary

IV. Result

The ROUGE score results for the Text − To − Text Transfer Transformer (T5) model is present in Table II.

Table 1: Result on the Dataset using T5-Base Model

	ROUGE-N	ROUGE-L
F1-Score	0.64	0.67
Precision	0.89	0.83
Recall	0.56	0.57

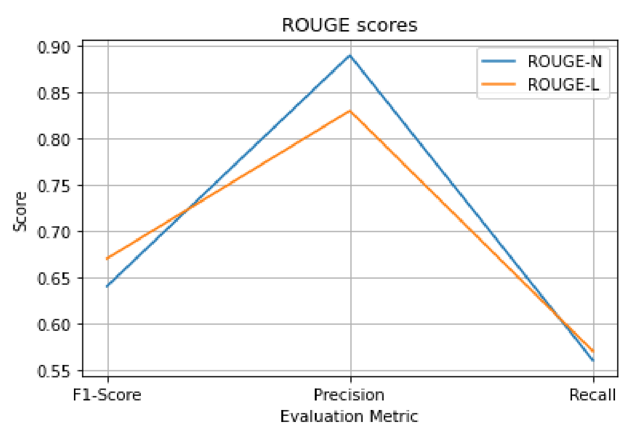

Figure 4: ROUGE Score

Figure 4 represents the ROUGE – L and ROUGE – N Score for the dataset with all the parameters such as Recall, Precision and F1-Score.

V. Conclusion

In conclusion of this research on abstractive text summarization, it highlights the key insights and contributions that have emerged from our research. We found that some models work better with abstractive summarization and some with extractive summarization. We also found that the transformer architecture is the best architecture for text summarization problems as it solves the problem of long-range dependencies and parallelization which is present in RNNs and LSTMs.

Furthermore, the research shows that this text summarization model works best on non-conversational text input.

VI. Future Work

A lot of promising work is to be done in the field of text summarization. This means inventing new architectures or using transfer learning to allow models to perform better in varied contexts.

The following scenarios could be done-

- Multi-document summarization could be done to combine multiple documents and provide a cohesive summary.
- We can explore methods which can help us control the level of abstraction in the summary.
- User centered approaches for summarization that considers user preference, need and context.

References

[1] Garg, K.D., Khullar, V. and Agarwal, A.K., 2021, August. Unsupervised machine learning approach for extractive Punjabi text summarization. In 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 750-754). IEEE.

[2] Teja, K.M., Sai, S.M. and Kushagra, P.S., 2018, November. Smart Summarizer for Blind People. In 2018 3rd International Conference on Inventive Computation Technologies (ICICT) (pp. 15-18). IEEE.

[3] Taeho Jo, "K Nearest Neighbor for Text Summarization using Feature Similarity." International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017.

[4] N. Moratanch, Dr. S. Chitrakala, "A Survey on Abstractive Text Summarization." International Conference on Circuit, Powe and Computing Technologies (ICCPCT), 2016.

[5] Gupta, P., Tiwari, R. and Robert, N., 2016, April. Sentiment analysis and text summarization of online reviews: A survey. In 2016 International Conference on Communication and Signal Processing (ICCSP) (pp. 0241-0245). IEEE.

[6] Dharmendra Hingu, Deep Shah, Sandeep S.Udmale, "Automatic Conference on Communication, Information & Computing Technology (ICCICT), 2015.

[7] Prakash, C. and Shukla, A., "Human aided text summarizer saar using reinforcement learning" 2014 International Conference on Soft Computing and Machine Intelligence, pp. 83-87, 2014.

[8] Kavita Ganesan, ChengXiangZhai, Jiawei Han, "Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions." Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 340 348, 2010.

[9] Chin-Yew Lin, "Rouge: A Package for Automatic Evaluation of Summaries." Barcelona Spain, Workshop o Text Summarization Branches Out, Post- Conference Workshop of ACL 2004.

[10] S. Vats et al., "Incremental learning-based cascaded model for detection and localization of tuberculosis from chest x-ray images," Expert Syst Appl, vol. 238, p. 122129, Mar. 2024, doi: 10.1016/J.ESWA.2023.122129.

[11] V. Sharma et al., "OGAS: Omni-directional Glider Assisted Scheme for autonomous deployment of sensor nodes in open area wireless sensor network," ISA Trans., Aug. 2022, doi: 10.1016/j.isatra.2022.08.001.