**Transformer Based Text Summary Generator**

*Pramod Kanjalkar, Rashmi Asolekar, Niraj Bhoi ,Purva Gharat,*

*Department of Instrumentation & Control Engineering*

*Vishwakarma Institute of Technology, Pune, India*

***Abstract—***

***In today’s fast-paced world, people find it difficult to read complete articles due to time constraints. To overcome this problem, the proposed system aims to develop an end-to-end deep learning application wherein users can give any article/new/research paper as input to get a precise summary of it, while conserving the overall meaning of the complete article. Extractive summarization and abstractive summarization are two separate methods of generating summaries. The extractive technique identifies the relevant sentences from the original document and extracts only those from the text. Whereas in abstractive summarization techniques, the summary is generated after interpreting the original text, hence making it more complicated. In this paper, we will be presenting a transformer model for abstractive text summarization. We have used BART stands for Bidirectional and Auto Regressive Transformers[5]. It is built with a seq2seq model trained with denoising as a pre-training purpose.***

***Keywords- Deep Learning; Natural Language Processing; Transformer; BART.***

1. INTRODUCTION

The amount of text data available online is increasing at a very fast pace hence text summarization has become essential. Most of the modern recommender and text classification systems require going through a huge amount of data. Manually generating precise and fluent summaries of lengthy articles is a very tiresome and time-consuming task. Hence generating automated summaries for the data and using it to train machine learning models will make these models space and time-efficient.

Text summarization can be broadly classified into two approaches -

Extractive Summarization - In extractive summarization, a summary from the given text is created by selecting a subset of the total sentence base. Most important phrases or sentences from the text are identified and selected based on a score that is computed depending on the words in that sentence.

Abstractive Summarization - In the method of abstractive summarization, an interpretation is first created by analyzing the text document. Based on this interpretation, the machine predicts a summary. It transforms the text by paraphrasing sections of the original document.

This work will focus on abstractive summarization to create an accurate and fluent summary as this task is more challenging and simulates human perception for developing summaries. For this task, we have used some machine learning models pretrained on a large dataset.

The following is a breakdown of the paper's structure. The second section of the paper presents a literature survey. The algorithm and proposed system are explained in Section III and IV respectively. The outcomes of the experiments are discussed in Section V. The paper comes to a close with Section VI.

1. LITERATURE SURVEY

The paper ‘Attention Is All You Need’ describes transformers and what is called a sequence-to-sequence architecture. Sequence-to-Sequence (or Seq2Seq) is a neural network that transforms a given sequence of elements, such as the sequence of words in a sentence, into another sequence[1] . The proposed methodology in the ‘Text Summarization using Neural Networks’ paper focuses only on the relevant sentences and passes them to the Bi-Directional RNN for identifying and representing the core idea of the article. Sequence to Sequence encoder-decoder model (RNN) has been used. It is more accurate than classical machine learning and statistical techniques. The drawback is that still struggles with much larger documents because of vanishing gradient problems in RNNs[2] .

In 2019 Chintan Shah and Anjali Jivani published a research paper named ‘An Automatic Text Summarization on Naive Bayes Classifier Using Latent Semantic Analysis’, The model that they have developed uses Naïve Bayes to distinguish whether sentences are likely to be extracted or not. It is a trainable and learnable summarization but can only perform extractive summarization and struggles with larger documents[3] . The proposed methodology in the ‘Automated News Summarization Using Transformers’ paper is that they have given a comprehensive comparison of a few transformer architecture-based pre-trained models for text summarization. They implemented pre-trained language models, which were based upon the transformer architecture for the task of summarization. They concluded from their research that finely tuned transformers based on pre-trained language models gave wonderful results and created a sound and fluent summary for a given text document[4] .

1. ALGORITHM

The transformer network[1] is solely based upon multiple attention layers. It does not make use of RNN and is reliant on attention layers and positional encoding for remembering the sequence of words in the input sequence. The global dependencies created with the help of multiple attention layers help in creating parallelization in processing the input.

The transformer model[1] contains encoder and decoder layers, where each is connected to a multi-head attention layer and feed forward network layers. The model remembers the position and sequence of words with the help of cosine and sine functions that creates positional encoding. The multi-head attention layer [1] in the encoder and decoder layer applies a mechanism called self-attention. The input is fed into three connected layers to create query (Q), key (K), and value (V) vectors [1].

These vectors are split into n vectors.

Self-attention is applied on n separate vectors to create multi-head attention[1].

where,

ℎ𝑒𝑎𝑑𝑖 = 𝐴𝑡𝑡𝑒𝑛𝑡𝑖𝑜𝑛(𝑄𝑊𝑄i,𝐾𝑊𝐾i , 𝑉𝑊𝑉i )

Where the projections are parameter matrices [1]

𝑊𝑄i ∈ ℝ𝑑𝑚𝑜𝑑𝑒𝑙×𝑑𝑘, 𝑊𝐾i ∈ ℝ 𝑑𝑚𝑜𝑑𝑒𝑙×𝑑𝑘, 𝑊𝑉i ∈ ℝ𝑑𝑚𝑜𝑑𝑒𝑙×𝑑𝑉 and 𝑊𝑂 ∈ ℝℎ𝑑𝑉×𝑑𝑚𝑜𝑑𝑒𝑙

Figure 1 [1] depicts the architecture of a transformer model. It contains an encoder and decoder layer and the various normalization and multi-head attention layers are also depicted in the figure.

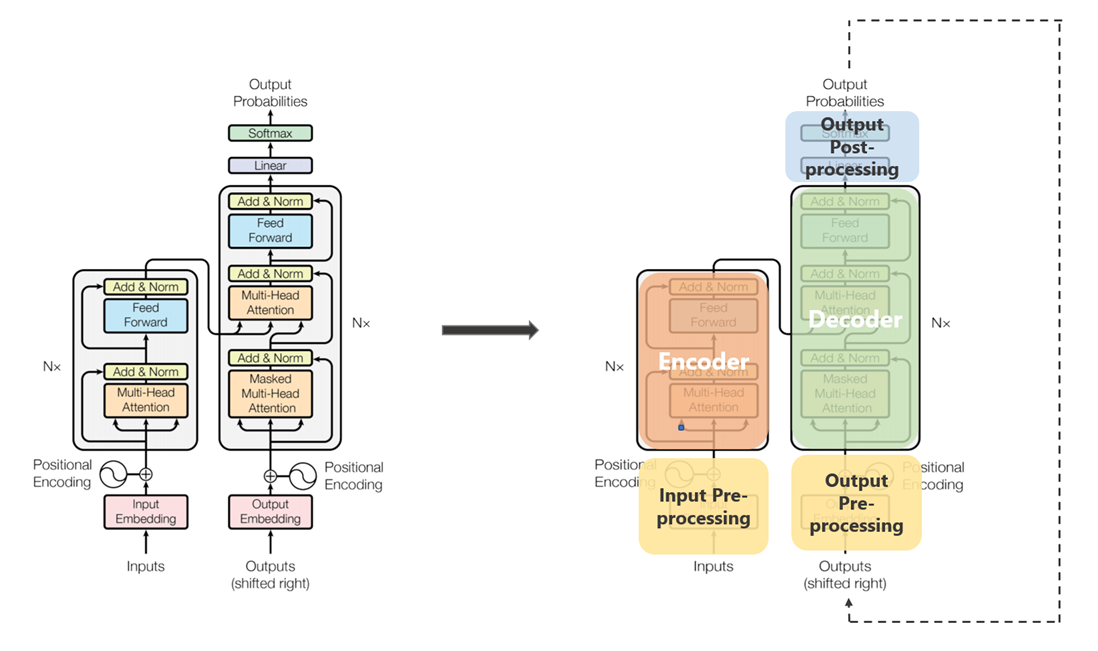


Fig. 1. Transformer Model Architecture [1]

**Pretrained Models based on Transformers -** Hugging face works as an open source for providing many useful NLP libraries and datasets. Its most famous library is the Transformer library. The transformer library consists of various pre-trained models to predict summaries of texts that can be fine-tuned for any dataset. The models we used are as follows:

**Pipeline –** The pipelines are a great and quick way to use different pre-trained models for inference. These pipelines are objects that abstract most of the library's complicated code, offering a simple API dedicated to several tasks, including text summarization. Pipelines enclose the overall steps of every NLP process such as Tokenization, Inference, which maps every token into a more meaningful representation, and Decoding.

**BART –** BART stands for Bidirectional and Auto Regressive Transformers[5]. It is built with a seq2seq model trained with denoising as a pre-training purpose. It uses a standard seq2seq model architecture combining an encoder similar to BERT and a GPT-like decoder. The pre-training task involves changing the order of the original phrases randomly and a new scheme where text ranges are switched with a single mask token. The large model of BART[5] consists of twice as many layers as are present in the base model. It is quite similar to the BERT model but BART contains about 10% more features than the BERT model of comparable size. BART's decoder is autoregressive, and it is regulated for generating sequential NLP tasks such as text summarization. The data is taken from the input but changed, which is closely related to the denoising pre-training objective. Hence, the input sequence embedding is the input of the encoder, and the decoder autoregressive produces output. We have used the "facebook/bart-large-cnn'' pre-trained model and then the Bart tokenizer, which is constructed from the GPT-2 tokenizer. Hence words are encoded differently depending on their position in the sentence.

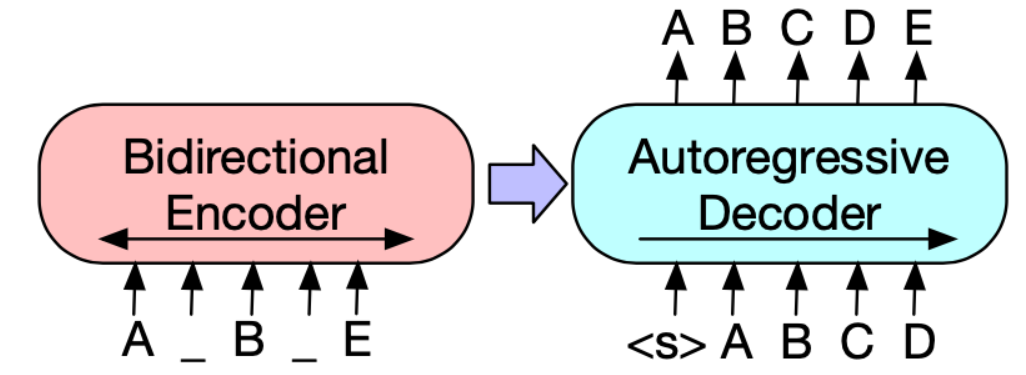


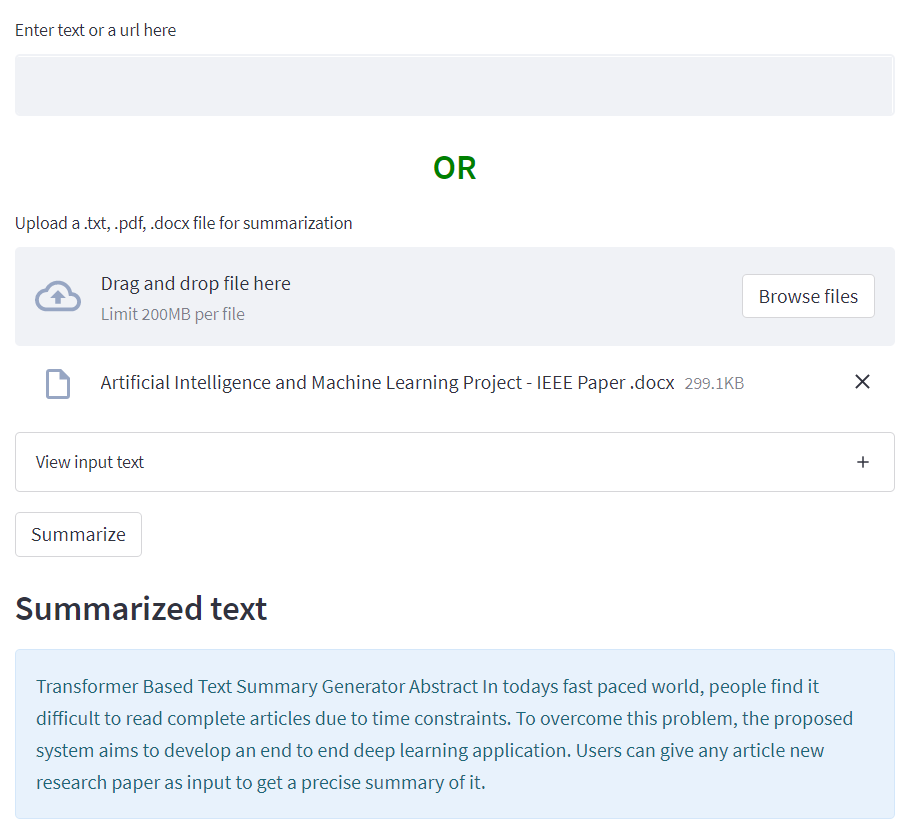
Fig. 2. BART Architecture [5]

1. EXPERIMENTATION

The text transformer algorithm is used for the summarization of text given as input by user. The method used for transformer is Bi-directional and Auto-Regressive Transformer (BART). The BART algorithm used for summarization is developed by Facebook. IT is encoder-encoder or seq2seq model which is similar to BERT (Bi-directional Encoder). For the purpose of summarization, various steps are concerned. First step is creating a web app for the user interface, where user can give different inputs to algorithm. Next step is the pre-processing of text (data) which is taken as input from user. This pre-processed data is then given to the model for further computations. After the data is passed through the model, then the summarized text is shown in the web app. The steps involved in experimentation are explained hereafter.

**Creating Web App**The web app was created for this experimentation using Streamlit. Streamlit is free and open-source framework to create web apps. Streamlit framework is based on Python Programming Language. This framework is specially designed for Data Scientists and Machine Learning enthusiasts for easily creating web apps for their projects without any dependency on web developers. Streamlit does not require any skills of web developer such as HTML, JavaScript, CSS, etc. This framework is compatible with most of the machine learning libraries of Python such as pandas, numpy, keras, tensorflow, matplotlib, PyTorch, etc. The framework speeds up the computational pipelines of transformers since it has functionality of data caching.

The web app created for this experimentation has functionalities for giving input as text, URL and PDF or document files. PDF or document files can be browsed by the user. Input data can be taken from any of this sources for further computations. A button is provided to get summarized text for given input text. The summarized text is shown on the web app in textbox. This basic web app is developed for experimentation. Fig.3 and shows the images of the web app.

Fig. 3. Web App Experimentation

**Pre-Processing on Input Data**The input data is taken from the user in different formats such as text, URL’s and PDF or document files. This input data is then cleaned using different pre-processing techniques. The cleaned data is then used by the summarizer model to summarize the given input. The pre-processing removes the noisy and irrelevant components from the data. These components can be of different types such as emoticons, symbols, images, URL’s, hashtags, mentions, extra spaces, special characters, etc. Emoticons are emoji that are widely used in social media. Hashtags and mentions are part of captions and comments on social media. Sometimes there can be extra spaces in the text, which are needed to be removed. The removal of this irrelevant data is done using Regular Expressions (Regex). Regex is the functionality of NumPy library of Python. If the data is large, it is divided into chunks. The chunks are then passed to the summarization model for further processing. The limit for one chunk or one part is set to 500 words. If the data exceeds 500 words, data is split into chunks.  
If the input data is given as text, data is first pre-processed as mentioned above. Chunks are created if required and then the data is sent to model for further processing. If the input data is taken in form of a URL, firstly the URL is validated using a readymade function in Python. Then the data is pre-processed, divided into chunks, and processed further. If input data is taken in form of a PDF or document file, the data is extracted from the file and then pre-processed, divided into chunks and processed further.

**BART Summarizer**The pre-processed data is sent to the BART model for summarization. In this stage, the data is first tokenized using the tokenizer. Then the tokenized data is given to the BART model and summarization is done. BART model uses encoding and decoding of data.

1. RESULT AND DISCUSSION

BART gave better results than the pre-trained Bart model in the pipeline method. The summaries generated were fluent, accurate, and integrated supporting evidence from the input document. Hence, the summaries generated by the BART pre-trained model demonstrate that the BART model is useful for text understanding.

1. CONCLUSION

For generating the summary of the given text of the document, a transformer-based approach was used. It was a pre-trained model of BART. BART is the seq2seq model for the summarization of text. Experimentation was carried out on different types of data and it was found to be performed satisfactorily.

1. REFERENCES

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30.

<https://arxiv.org/abs/1706.03762>

[2] A. Jadhav, R. Jain, S. Fernandes and S. Shaikh, "Text Summarization using Neural Networks," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019, pp. 1-6, doi: 10.1109/ICAC347590.2019.9036739.

<https://ieeexplore.ieee.org/document/9036739>

[3] Shah C, Jivani A. An Automatic Text Summarization on Naive Bayes Classifier Using Latent Semantic Analysis. InData, Engineering and Applications 2019 (pp. 171-180). Springer, Singapore.

<https://link.springer.com/chapter/10.1007/978-981-13-6347-4_16>

[4] Gupta A, Chugh D, Katarya R. Automated news summarization using transformers. InSustainable Advanced Computing 2022 (pp. 249-259). Springer, Singapore.

<https://link.springer.com/chapter/10.1007/978-981-16-9012-9_21>

[5] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. CoRR abs/1910.13461:

<https://arxiv.org/abs/1910.13461>