International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5902
***Article Received**: 14 October 2022 **Revised**: 10 November 2022 **Accepted**: 05 December 2022*

Fine Tuning Transformer Based BERT Model for Generating the Automatic Book Summary

Prottyee Howlader[1], Prapti Paul[2], Meghana Madavi[3], Dr. Laxmi Bewoor[*4,] Dr. V.S.Deshpande[5]
[1] *Vishwakarma Institute of Information Technology, Pune-411048,INDIA*
ORCID ID : 0000-0003-4470-0239
[2] *Vishwakarma Institute of Information Technology, Pune-411048,INDIA*
ORCID ID : 0000-0002-2312-8600
[3] *Vishwakarma Institute of Information Technology, Pune-411048,INDIA*
ORCID ID : 0000-0003-4491-3234
[4]*Vishwakarma Institute of Information Technology, Pune-411048,INDIA*
ORCID ID : 0000-0003-3470-421X
** Corresponding Author Email: laxmi.bewoor@viit.ac.in*

Abstract:
Major text summarization research is mainly focusing on summarizing short documents and very few works is witnessed for long document summarization. Additionally, extractive summarization is more addressed as compared with abstractive summarization. Abstractive summarization, unlike extractive summarization, does not only copy essential words from the original text but requires paraphrasing to get close to human generated summary. The machine learning, deep learning models are adapted to contemporary pre-trained models like transformers. Transformer based Language models gaining a lot of attention because of self-supervised training while fine-tuning for Natural Language Processing (NLP) downstream task like text summarization. The proposed work is an attempt to investigate the use of transformers for abstraction. The proposed work is tested for book especially as a long document for evaluating the performance of the model.

Keywords: Summarization, Abstractive, Extractive, Transformers.

I. Introduction:

The digitalization era led to abundant information majorly available in unstructured way. Though it's difficult and time consuming to get relevant and necessary information but simultaneously provides an opportunity to get insight for handling the challenges of information retrieval. Summarization is one of the critical and important task as it is a technique for condensing a lengthy source text into a brief summary that captures the main points and these summaries not only saves the time but also considered for knowing the relevance and need .

Extractive and abstractive techniques are the two most prominent paradigms in document summarization [1]. Extractive procedures build summaries by extracting elements of the original content (generally sentences) whereas abstractive summarizers get its name from the fact that they don't employ sentences from the original text passage to construct the summary. Instead, they create a paraphrase of the provided text's major points, using a vocabulary set that differs from the original. Abstractive methods are challenging as it produce new words or phrases that are not in the original material [2]. The efforts were made in developing various machine learning and deep learning models. Deep learning models found to be more efficient for abstractive summarization [3]. Nowadays, transformer based architecture is stepping forward for developing efficient text summarization techniques [4-5]. The development of transformer and its subsequent enhancements progressed to models like Bidirectional Encoder Representation from Transformers (BERT) which ultimately proving to be a better solution for Natural Language Processing (NLP) models [6].

This paper presents abstractive text summarization technique using BERT for book summarization. The paper is organized as follows: Section 2 describes related work for abstractive summarization. Section 3 describes BERT architecture .Section 4 describes BERT models. Section 5 elaborates BERT implementation and evaluation metric for book summarization. Section 6 concludes the paper.

Related Work:

This section is focusing on early work on abstraction as well as on-going development in this domain. Andhale and Bewoor[7] reviewed various approaches to deal with text summarization. Recently Deep Learning gained popularity for handling abstractive summarization challenges [3]. In spite the fact of usefulness of deep learning models like Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) for Natural Language Processing (NLP) tasks still has the limitation of vanishing gradient and slower training. Nowadays Transformer based architecture is providing a strong insight for NLP downstream tasks [4-5]. This section is providing major contribution by transformers for text summarization.

During the last two decades, automatic extraction text summary on lectures has shown to be a useful method for extracting essential words and sentences that best reflect the content. Many existing procedures, on the other hand, use obsolete methodologies, resulting in mediocre outputs or requiring hours of human tweaking to get meaningful results. New machine learning architectures have recently presented strategies for extractive summarization by clustering of deep learning model output embedding. The proposed work used the "lecture summarising service," a RESTful Python service that employed the BERT model for text embedding and K-Means clustering to find phrases closest to the centroid for summary selection. The purpose of the service was to provide students with a tool that would allow them to summarise course content in any number of phrases they desired. In addition to summary work, the service offered lecture and summary management, as well as information storage in the cloud. While the results of employing BERT for extractive text summarization were promising [8].

Liu and Lapata[~~9~~ 8] applied BERT for summarizing text and proposed a common framework for extractive and abstractive models. A novel document-level encoder based on BERT was introduced for providing the semantics of a document and in order to get representations for the sentences. Several inter-sentence transformer layers were stacked on top of the encoder for building the extractive model. The fine tuning schedule was proposed for abstractive summarization for performance improvement. In order to evaluate performance of transformers for language generation, Zhang et al.[12] proposed BERT based natural language generation model. The pre-trained language model of encoder and decoder was used for training the model. The word from the summary was generated by considering context from both sides. The attempt was made for use of BERT for extraction first and eventually use extracted text for abstraction. Recently, Ma et al.[10] proposed T-BERTSum model which wais a blend of topic aware extractive and abstractive summarization model . The model simultaneously was providing inference from topics and later producing summarization from social texts. The model was primarily focusing on long-term dependencies which was trained through the transformer network .The use of Long Term short-term memory (LSTM) network layers on the top of the extractive model were filtering the effective information for building abstractive model. The use of BERT was tested for different languages other than English.

Elmadani et.al[11] applied multilingual BERT for summarizing Arabic text. This was first attempt for using BERT other than English language.

Iwasaki et al.[13] developed algorithm in Japanese using a neural network for abstractive text summarization. The feature-based input vector of sentences was gained by encoder with the use of BERT. The summary sentence was the output from the transformer-based decoder

Here, we are using BERT for abstractive document summarization. BERT stands for Bidirectional Encoder Representations from Transformers. BERT is a specific, large transformer-masked language model that has the capacity to learn specific tasks when it comes to language and it can do so because it has the understanding of how words relate to each other for which, it can perform various tasks when it is fine-tuned. The model is trained in two separate chunks which are pre-training and fine-tuning respectively for performing various tasks.

II. Comparison of GPT-Neo and BERT for long text summaries

Humans continuously require benchmarks to determine which one performs better in various systems to construct general-purpose AI systems that can cope with uncertainty across new areas. In almost every NLP challenge, pre-trained language models based on transformers have done well. These models provide essential baseline knowledge for assembly processing, eliminating the requirement for subsequent models to be trained from the bottom up. GPT-Neo and BERT are two neural network and deep learning-based approaches for building language models. These are relatively new in the domain of natural language processing, yet they outperform practically every other concept. In this experiment, we compare the two conceptual strategies to evaluate which one is better for text summary.

2.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a pre-train deep bidirectional representation of unlabelled text trained in left and right context in all layers. Further, the model is fine-tuned merely with output layer to provide cutting-edge models without bearing major task specific architectural changes for a varied tasks like question answering or language inference [14]. BERT may be trained for representation learning in an unsupervised manner, and then fine-tuned in a supervised manner on the so-called downstream tasks [4]. There are pre-trained versions of BERT that may be fine-tuned and utilized to solve unique supervised learning assignment. With this TensorFlow lesson, users may use a pre-trained BERT model. And, the original transformation was not intended to be a language

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5902
***Article Received**: 14 October 2022 **Revised**: 10 November 2022 **Accepted**: 05 December 2022*

model, but rather to solve sequence transduction problems simply using attention rather than recurrent connections. The original transformer had both an encoder and a decoder, whereas BERT only has an encoder. Because it employs an encoder that is quite close to the transformer's original encoder, BERT is a transformer-based model.

2.2 GPT-Neo (Generative Pre-Trained Transformers)

GPT-Neo is a set of codebases for training big language models that is being released as open source. GPT-neo is mostly used to create blog articles and music. The sizes of the models are being used to refer to them. GPT-Neo 1.3B is a transformer model based on EleutherAI's GPT-3 architectural replication. EleutherAI is an open-source community artificial intelligence project with the goal of creating a fully distributed singleton AI with its own decentralized civilization. GPT-Neo denotes the kind of model, whereas 1.3B is the number of parameters in this pre-trained model. EleutherAI created a large-scale curated dataset that was used to train GPT-Neo 1.3B.This model was trained for 380 billion tokens across 362,000 steps on the Pile. It was trained using cross-entropy loss as a masked autoregressive language model. The model learns an inner representation of the English language in this manner, which it may then use to extract characteristics helpful for downstream applications. However, the model excels at what it was trained for: creating sentences given a prompt.

As the main intuition for the proposed work was to generate abstractive text for lengthy document the input considered was a book. The general flow followed for the models were:

$$\begin{tikzcd}\boxed{\text{Install transformer}} \arrow[r] & \boxed{\text{Import pipeline for the model}} \arrow[r] & \boxed{\text{Input the text file}} \arrow[r] & \boxed{\text{Output}}\end{tikzcd}$$

Fig 1: General Flow for GPT-Neo and BERT

III. Methodology

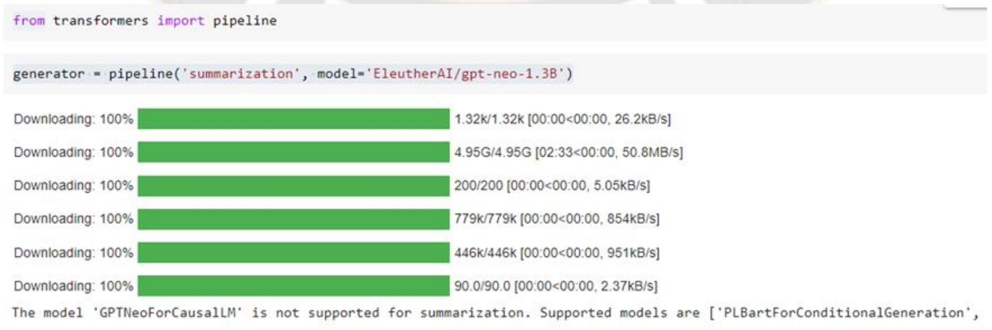
The model takes a Portable Document Format (PDF) file as an input. The text cleaning and removal of stop-words was done and the cleaned text was then passed to the Transformer model for summarization. Additionally text wrapping was also considered because it helps in preserving new lines, spaces and tabs. It is also used to prevent horizontal scrolling. The step found to be essential as it is wrapping the output to avoid any horizontal scroll bars.

As the main objective of the proposed work is to build a model which can generate summaries for long text we have inputted book titled “Rich Dad Poor Dad” by Robert T Kiyosaki which contains 9 Chapters across 234 Pages having a total word count of 55,000 words. The input was given to two models viz. GPT-Neo and BERT.

Transformers provide APIs that make it simple to download and train cutting-edge pre-trained models. The models may be applied to a variety of tasks, including text categorization, summarization, and text production. Here we are using the hugging face transformers library. We are installing the transformers along with sentencepiece as some of the tokenizers need sentencepiece. SentencePiece is a Neural Text Processing sub word tokenizer and de-tokenizer that consider the input as a raw input stream, along with including the space in the set of characters to use.

3.1 GPT-Neo

The general flow as depicted in Fig.1 is followed for GPT_Neo.



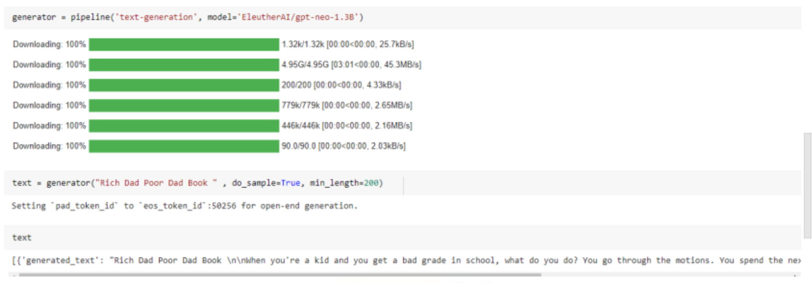
```
from transformers import pipeline

generator = pipeline('summarization', model='EleutherAI/gpt-neo-1.3B')

Downloading: 100%  1.32k/1.32k [00:00<00:00, 26.2kB/s]
Downloading: 100%  4.95G/4.95G [02:33<00:00, 50.8MB/s]
Downloading: 100%  200/200 [00:00<00:00, 5.05kB/s]
Downloading: 100%  779k/779k [00:00<00:00, 854kB/s]
Downloading: 100%  446k/446k [00:00<00:00, 951kB/s]
Downloading: 100%  90.0/90.0 [00:00<00:00, 2.37kB/s]
The model 'GPTNeoForCausalLM' is not supported for summarization. Supported models are ['PLBartForConditionalGeneration',
```

Fig.2 : GPT_Neo model

The transformer’s library after successful installation pipelines was installed. The text file was given as input to GPT_Neo for summary generation.However, it demonstrates that GPT-Neo is unable to provide summarization. It conveys the following message: “**The model 'GPTNeoForCausalLM' is not supported for summarization”.** GPT-Neo was essentially trained as an autoregressive language model. This means that its main job

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5902
***Article Received**: 14 October 2022 **Revised**: 10 November 2022 **Accepted**: 05 December 2022*

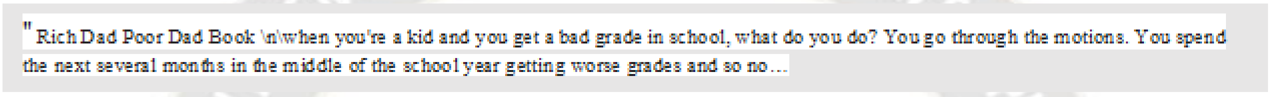
is to anticipate the next character from a string of text. As a result, it is ideal for text generation but not for summarization. This was proven when we attempted to generate the text for the book Rich Dad Poor Dad.



```
generator = pipeline('text-generation', model='EleutherAI/gpt-neo-1.3B')

Downloading: 100% 1.32k/1.32k [00:00<00:00, 25.7kB/s]
Downloading: 100% 4.95G/4.95G [03:01<00:00, 45.3MB/s]
Downloading: 100% 200/200 [00:00<00:00, 4.33kB/s]
Downloading: 100% 779k/779k [00:00<00:00, 2.65MB/s]
Downloading: 100% 446k/446k [00:00<00:00, 2.16MB/s]
Downloading: 100% 90.0/90.0 [00:00<00:00, 2.03kB/s]

text = generator("Rich Dad Poor Dad Book " , do_sample=True, min_length=200)

Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

text

[{'generated_text': "Rich Dad Poor Dad Book \n\nWhen you're a kid and you get a bad grade in school, what do you do? You go through the motions. You spend the ne
```

Fig.3 : Text Generation by GPT_Neo model

After installing the pipelines, GPT_Neo was directed to generate the text to check the relevance for generating the abstract. However following text presented in Fig.4 was generated by the model:



" Rich Dad Poor Dad Book \n\nwhen you're a kid and you get a bad grade in school, what do you do? You go through the motions. You spend the next several months in the middle of the school year getting worse grades and so no…

Fig.4 : Abstract generated by GPT_Neo model

The output generated by the model was not relevant to the abstract expected and hence it has been decided not to use this model for abstraction.

3.2 BERT

The general flow as depicted in Fig. 5 is followed for BERT model.

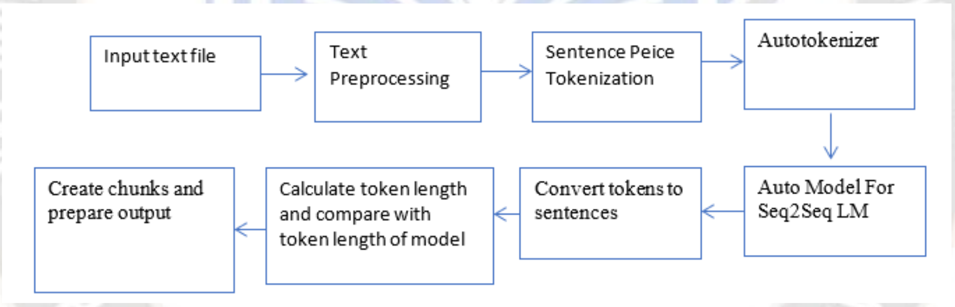

Input text file
Text Preprocessing
Sentence Peice Tokenization
Autotokenizer
Auto Model For Seq2Seq LM
Convert tokens to sentences
Calculate token length and compare with token length of model
Create chunks and prepare output

Fig.5 : Process Flow by BERT model

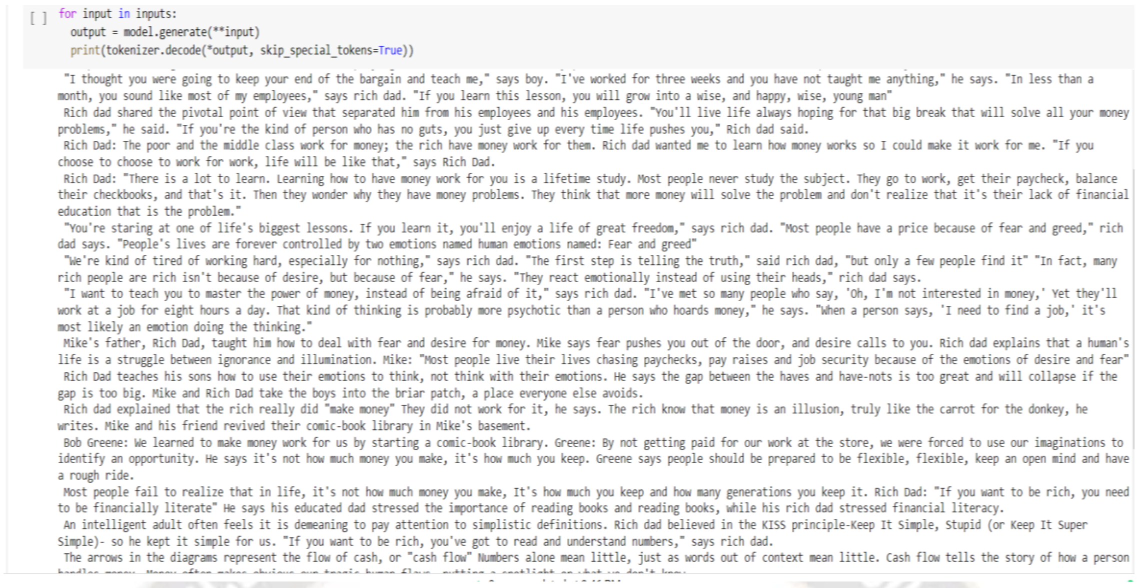
BERT consists of two steps: Pre-training and Fine-tuning. Unlabeled data is used to train the model for various tasks, during the pre-training stage. The BERT model is fine-tuned on labeled data from downstream tasks after it has been initialized with the pre-trained parameters.

The process begins with the same steps shown in Fig. 1. Sentencepiece Tokenization is a sub word tokenization algorithm used by BERT which was installed along with the transformers. SentencePiece is a Neural Text Processing sub word tokenizer and de-tokenizer that consider the input as a raw input stream, along with including the space in the set of characters to use.

AutoTokenizer and AutoModelForSeq2SeqLM from the transformers library were imported. AutoTokenizer creates a class of the relevant architecture when it is instantiated. When using the AutoTokenizer from pretrained() class function, a generic tokenizer will be instantiated as one of the library's tokenizer classes. Similarly, we will be initializing the model using AutoModelForSeq2SeqLM for fine tuning the model.

Later, the number of tokens including the special tokens was set for the model. The file contents were split into sentences using Natural Language Toolkit (NLTK) library. Similarly, the maximum number of tokens present in the longest sentence of the file content was calculated and checked whether the number of tokens present in the longest sentence is below the limit of the maximum number of tokens supported by the model.

For getting the final output we need to decode the output by skipping the special tokens. After running the model, following final summary of the text file was generated.



```
[ ] for input in inputs:
      output = model.generate(**input)
      print(tokenizer.decode(*output, skip_special_tokens=True))
```

```
 "I thought you were going to keep your end of the bargain and teach me," says boy. "I've worked for three weeks and you have not taught me anything," he says. "In less than a
month, you sound like most of my employees," says rich dad. "If you learn this lesson, you will grow into a wise, and happy, wise, young man"
 Rich dad shared the pivotal point of view that separated him from his employees and his employees. "You'll live life always hoping for that big break that will solve all your money
problems," he said. "If you're the kind of person who has no guts, you just give up every time life pushes you," Rich dad said.
 Rich Dad: The poor and the middle class work for money; the rich have money work for them. Rich dad wanted me to learn how money works so I could make it work for me. "If you
choose to choose to work for work, life will be like that," says Rich Dad.
 Rich Dad: "There is a lot to learn. Learning how to have money work for you is a lifetime study. Most people never study the subject. They go to work, get their paycheck, balance
their checkbooks, and that's it. Then they wonder why they have money problems. They think that more money will solve the problem and don't realize that it's their lack of financial
education that is the problem."
 "You're staring at one of life's biggest lessons. If you learn it, you'll enjoy a life of great freedom," says rich dad. "Most people have a price because of fear and greed," rich
dad says. "People's lives are forever controlled by two emotions named human emotions named: Fear and greed"
 "We're kind of tired of working hard, especially for nothing," says rich dad. "The first step is telling the truth," said rich dad, "but only a few people find it" "In fact, many
rich people are rich isn't because of desire, but because of fear," he says. "They react emotionally instead of using their heads," rich dad says.
 "I want to teach you to master the power of money, instead of being afraid of it," says rich dad. "I've met so many people who say, 'Oh, I'm not interested in money,' Yet they'll
work at a job for eight hours a day. That kind of thinking is probably more psychotic than a person who hoards money," he says. "When a person says, 'I need to find a job,' it's
most likely an emotion doing the thinking."
 Mike's father, Rich Dad, taught him how to deal with fear and desire for money. Mike says fear pushes you out of the door, and desire calls to you. Rich dad explains that a human's
life is a struggle between ignorance and illumination. Mike: "Most people live their lives chasing paychecks, pay raises and job security because of the emotions of desire and fear"
 Rich Dad teaches his sons how to use their emotions to think, not think with their emotions. He says the gap between the haves and have-nots is too great and will collapse if the
gap is too big. Mike and Rich Dad take the boys into the briar patch, a place everyone else avoids.
 Rich dad explained that the rich really did "make money" They did not work for it, he says. The rich know that money is an illusion, truly like the carrot for the donkey, he
writes. Mike and his friend revived their comic-book library in Mike's basement.
 Bob Greene: We learned to make money work for us by starting a comic-book library. Greene: By not getting paid for our work at the store, we were forced to use our imaginations to
identify an opportunity. He says it's not how much money you make, it's how much you keep. Greene says people should be prepared to be flexible, flexible, keep an open mind and have
a rough ride.
 Most people fail to realize that in life, it's not how much money you make, It's how much you keep and how many generations you keep it. Rich Dad: "If you want to be rich, you need
to be financially literate" He says his educated dad stressed the importance of reading books and reading books, while his rich dad stressed financial literacy.
 An intelligent adult often feels it is demeaning to pay attention to simplistic definitions. Rich dad believed in the KISS principle-Keep It Simple, Stupid (or Keep It Super
Simple)- so he kept it simple for us. "If you want to be rich, you've got to read and understand numbers," says rich dad.
 The arrows in the diagrams represent the flow of cash, or "cash flow" Numbers alone mean little, just as words out of context mean little. Cash flow tells the story of how a person
```

Fig: 6 Abstract by BERT model

IV. Results

The performance of model is quantified with evaluation metric. Normally Precision, Recall, F1 score evaluation metric is used in NLP context [].These evaluation metric is quite simple for extractive summarization as simply the extracted text is checked against available text. However these measures are difficult while it is used in abstraction. The book abstraction was very less attempted so for comparing the results there was no classical result available. Additionally, as abstract focuses on paraphrasing the output always differs. The proposed work is finding abstract for a book, the benchmark for comparison was considered as preface of the book as it provides gist of entire book. The findings for the work is described below.

We can't determine the accuracy, recall, or F1 score since GPT-Neo doesn't provide the summary. Indeed, we are grateful to OpenAI for GPT-Neo, which is a great alternative for text generation. However, because BERT provided the summary (abstract), we evaluated F1 score, recall and compared the findings to the real data set presented in the preface.

Data:

- Number of words that overlap between the " Preface of Rich Dad Poor Dad" book and the "summary generated by the system"**:** 2449
- In the preface, total number of words is presented as a summary of the book**:** 5775
- The total amount of words is generated by the system when the summary is performed using the BERT model**:** 4165

- **Recall** = $\frac{Number\ of\ overlap\ words}{The\ total\ number\ of\ words\ in\ the\ reference\ summary}$ (I)
  $= \frac{2449}{5775}$
  $= 0.424$
- **Precision** =$\frac{Number\ of\ overlap\ words}{The\ total\ number\ of\ words\ generated\ by\ the\ system}$ (II)
  $=\frac{2449}{4165}$
  $= 0.588$
- **F-Measure** =$\frac{Recall+Precision}{2}$ (III)
  $=\frac{0.424+0.588}{2}$
  $= 0.506$

The results obtained above clearly indicate that almost 50% system generated abstract overlap with the preface. The model needs to be improvised by providing a rich vocabulary in order to improve the text generation to cope with human cognitive level. The results are found to be promising but the results are not compared as no prior work found for this application.

V. Conclusion

GPT-Neo and BERT are two approaches for building language models based on machine learning and deep learning. They are both relatively new ideas in natural language processing. We examine the two conceptual processes in this experiment to see whether one is superior for lengthy text summarization**.** We discovered that when it comes to the issue of summarization for large file content, BERT is the best and most beneficial model after examining both GPT-Neo and BERT models. Finally, based on the results it is concluded that BERT condenses a large amount of data into a brief summary by providing the best wording possible.

References:

[1] N. S. Shirwandkar and S. Kulkarni, "Extractive Text Summarization Using Deep Learning," Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018, 2018, doi: 10.1109/ICCUBEA.2018.8697465.

[2] S. Singhal, "Abstractive Text Summarization," J. Xidian Univ., vol. 14, no. 6, pp. 1–11, 2020, doi: 10.37896/jxu14.6/094

[3] D. Suleiman and A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," Math. Probl. Eng., vol. 2020, 2020, doi: 10.1155/2020/9365340.

[4] A. Vaswani et al., "Attention is all you need," Adv. Neural Inf. Process. Syst., vol. 2017-December, no. Nips, pp. 5999–6009, 2017.

[5] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," IEEE Access, vol. 9, pp. 68675–68702, 2021, doi: 10.1109/ACCESS.2021.3077350.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[7] N. Andhale and L. A. Bewoor, "An overview of text summarization techniques," 2017, doi: 10.1109/ICCUBEA.2016.7860024.

[8] D. Hingu, D. Shah, and S. S. Udmale, "Automatic text summarization of Wikipedia articles," Proc. - 2015 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2015, pp. 15–18, 2015, doi: 10.1109/ICCICT.2015.7045732.

[9] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 3730–3740, 2019, doi: 10.18653/v1/d19-1387.

[10] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian and N. Al-Nabhan, "T-BERTSum: Topic-Aware Text Summarization Based on BERT," in *IEEE Transactions on Computational Social Systems*, vol. 9, no. 3, pp. 879-890, June 2022, doi: 10.1109/TCSS.2021.3088506.

[11] K. N. Elmadani, M. Elgezouli, and A. Showk, "BERT Fine-tuning For Arabic Text Summarization," pp. 2018–2021, 2020, [Online]. Available: http://arxiv.org/abs/2004.14135

[12] H. Zhang, J. Cai, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," CoNLL 2019 - 23rd Conf. Comput. Nat. Lang. Learn. Proc. Conf., pp. 789–797, 2019, doi: 10.18653/v1/k19-1074.

[13] Y. Iwasaki, A. Yamashita, Y. Konno and K. Matsubayashi, "Japanese abstractive text summarization using BERT," 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2019, pp. 1-5, doi: 10.1109/TAAI48200.2019.8959920.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for LanguageUnderstanding." https://arxiv.org/pdf/1810.04805.pdf