Text Summarization for Big Data Analytics: A Comprehensive Review of GPT 2 and BERT Approaches

G. Bharathi Mohan, R. Prasanna Kumar, Srinivasan Parathasarathy, S. Aravind, K. B. Hanish, and G. Pavithria

1 Introduction

At present, where everything is fast and simple for the convenience of humans, a short version of the existing long version is necessary.

People do not have the time and patience to consume information as a whole and they have to be fed in a simple manner. Professionals who handle large reports also face such problems. This can be solved by summarization of text. Text summarization is defined as

> Automatic text summarization is the process of compressing and extracting information effectively from input documents while still retaining its key content. [1]

A basic understanding on what a summary is required before moving on to the text summarization. A summary is a minimized version of something that is created from one or more texts, delivers the key ideas from the original text, and is written in a concise manner. Automatic text summarization aims to show the source text as a short version form with semantics [2]. The goal of text summarization is to come up with methods to produce this summary efficiently and clearly. There are two types of summarizers used in our case. They are abstractive and extractive summarizers.

By choosing a portion of the entire sentence base, extractive summarization creates a summary of the text that has been provided. The text's most significant phrases and sentences are determined and chosen with a score that is calculated based on the words in the text. The approach of abstractive summarization begins with analyzing the text document to develop an interpretation. The computer makes

G. Bharathi Mohan (✉) · R. Prasanna Kumar · S. Aravind · K. B. Hanish · G. Pavithria
Department of CSE, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India
e-mail: g_bharathimohan@ch.amrita.edu

S. Parathasarathy
Oracle America, Lehi, UT, USA

R. Sharma et al. (eds.), *Data Analytics for Internet of Things Infrastructure*,
Internet of Things, https://doi.org/10.1007/978-3-031-33808-3_14

a prediction based on this interpretation: a summary. By paraphrasing parts of the actual text, it changes the essence of the text [3].

Text summarization is classified into two types: indicative and informative. Inductive summary just provides the core concept of the text to the consumer. This form of summary is typically 5–10% of the length of the main portion of the text. Informative summary methods, on the other hand, give exact information on the primary text. The useful summary should be 20–30% as long as the main material [2].

At this stage, most of the automatic text summarization techniques mainly use machine learning or deep learning methodologies and models. Deep learning techniques were used for the first time in abstractive text summarization in 2015, with the proposed methodology based on the encoder-decoder architecture. Deep learning models have produced amazing results in these applications and have been widely used recently [4].

Our work compares two models of Transformer, GPT2 and BERT, to find out which one is more efficient than the other. At the end of our analysis using ROUGE metrics, it has been confirmed that BERT gives a better performance compared to GPT2.

2 Related Works

Subha Shini and Ambeth Kumar developed and put into use recurrent neural network-based text summarization techniques. They have addressed different approaches and distinct datasets that are used to create text summaries, which decrease the time required to manually enter summaries for lengthy texts while maintaining the meaning produced by recurrent neural networks (RNNs) [5].

Two of the LSA-based summarizing techniques are described in [6]. The algorithm results are compared using their ROUGE scores after being assessed on Turkish and English documents. Both of our systems perform equally well on collections of Turkish and English documents; however, one of them yields the highest scores.

The most recent extractive text summarizing methods for several languages have been described in this study [7]. It is seen that excellent work has been done for several foreign languages, like Chinese and English. However, there is still no summary system for Bengali languages. Therefore, it is difficult to provide a summary method that makes use of several feature categories.

In [8], the authors have designed and implemented a text summarization that uses the BERT algorithm. They introduced a document-level encoder based on BERT which expresses the document into segments and obtains a representation of the given paragraph. They also used a two-staged fine-tuning approach to further improve the quality of the summaries that is generated.

In another study, text summarization was accomplished using multi-layered attentional peephole convolutional LSTM (Long Short-Term Memory). The goal is

to create an input of a given lengthy text and construct an automatic text summation. They have improved the settings for MAPCol utilizing the central composite design (CCD) in conjunction with RSM, which produces summaries with good accuracy [9].

Jesse Vig introduced an open source tool as a way to visualize attention in the Transformer at various scales. The tool was demonstrated on GPT-2 and BERT, and three use cases were presented. To traverse the tool's three views, a uniform interface is intended to be created in future works [10].

A text summarization tool that uses the LSTM-CNN-based ATS framework (ATSDL) has been designed and implemented by some authors. The purpose is to construct new sentences which will explore more good-grained fragments than sentences generated. ATS is the task of forming summary sentences by combining facts from different sources and forming them into shorter content and preserving meaning [11].

In [12], the authors upgrade to a recurrent neural network architecture the most advanced model for abstractive sentence summarization. The model is a condensed form of the machine translation encoder-decoder structure. To produce headlines based on the first line of each news article, this model was trained on the Gigaword corpus. Gigaword data and the DUC-2004 challenge show that it performs far better than the prior state of the art, despite the fact that this model does not rely on any additional extraction properties.

The Improved Attention Layer-assisted-Recurrent Convolutional Neural network model is used for yet another text summarization (IA-RCNN). The Sequence-to-Sequence (S2S) paradigm was integrated with RCNN in the model, which was created for abstractive text summarization in a variety of text sources. The model that is suggested in this research is tested using various text sources. For real-time applications, its performance is adequate [13].

2.1 Text Summarization Using Deep Learning

In 2015, deep learning techniques were first used to summarize abstract text, and a model based on encoder/decoder architecture was proposed. Deep learning algorithms have become popular recently and have achieved excellent results in these applications.

Deep learning analyzes difficult situations as a decision aid. Deep learning uses feature extraction at different levels of abstraction to mimic the capabilities of the human brain. Deep learning is used in many NLP tasks because it uses multiple nonlinear layers of data processing to facilitate learning multi-level hierarchical representations of data. Various deep learning models were used for abstract summarization, including RNNs, convolutional neural networks (CNNs), and sequence-to-sequence models. In this section, deep learning models will be explored [27].

A large corpus is used to learn the contextual representations of the language. The BERT language representation is one of the new word embedding model extensions. BERT benefits from a fine-tuning and role-based approach (depending on specific job goals). In addition, transformers can learn the meaning of relationships between "word pairs" by using self-awareness to compute input and output representations [4].

2.2 Need for Text Summarization in Big Data Analytics

Big data is a technology that is used to manage massive amounts of data. It is a technique for storing, distributing, and processing massive amounts of data. Because of technology improvements and HTML 2.0, it is now possible to send data without specifying a tag. Specifically, social media tools such as WhatsApp, Facebook, Instagram, Google, and others are extremely effective for transmitting large amounts of organized and unstructured data [22]. When you view a video on YouTube, you will see advertising that are relevant to your interests at random intervals [23]. This advertising has been chosen for you based on your browsing history and preferences. The most current advancements in a certain issue may be found via Twitter trends.

Furthermore, Google text search and voice-based search have simplified searches nowadays. Today, most individuals use mobile phones to conduct various transactions, and these phones are also used to monitor user locations. Today, various mobile applications are available, and these programmers are installed based on the user's preferences and requirements [24]. These mobile applications are highly beneficial in understanding the user's interests and how much time they spend. These many scenarios suggest extensive data creation, with data being created in vast quantities at all times [25, 26]. As a result, a unique text summarizing approach is necessary to better grasp the hidden knowledge or information concealed behind this data.

3 BERT

BERT (Bidirectional Encoder Representations from Transformers) is used to get over the restriction of RNN (Recurrent Neural Networks), CNN (Convolution Neural Networks), and ANN (Artificial Neural Networks). BERT models are pre-trained on big data sets. So no additional training on the dataset is required. It uses an efficient uniform architecture along with sentence transformation in the resources. With this, the best results are observed in summarization. Unlike any context-free embedding model, BERT is a contextual embedding model [14]. BERT creates on top of the transformer architecture, but its purpose is distinct for pre-training data [15].

Example:
Consider these sentences:

Python is very dangerous. It can kill you.
I love Python. It's easy to understand.

If the word embedding is applied, it will consider the word python in two sentences together. This is not correct because one python is a snake and another is a programming language. So BERT helps us by keeping the contextual embedding of the model.

Advantages of the BERT model are as follows:

1. *Context* – It keeps the context of the words.
2. *Word Ordering* – It keeps the words ordering with the help of positional encoding or positional vectors.
3. *Embeddings* – It has token embedding, sentence embedding, and positional embedding. All these are BERT embedding.
4. *Out of Vocabulary* – It keeps out of vocabulary with the help of self-attention. Moreover, it takes care of those keywords, which are present or not present in the vocabulary.

Applications of the BERT model are as follows:

- Text Summarization
- NER – Name Entity Recognition
- Next Sentence Prediction
- Long Text Classification
- Sentiment Analysis
- Question Answering

3.1 BERT Architecture

The goal of Bidirectional Encoder Representations from Transformers (BERT), which pre-trains deep bidirectional representations from unlabeled text, is to simultaneously condition both left and right contexts across all layers. As a result, the pre-trained BERT model can be enhanced with just one additional output layer to create cutting-edge models for a variety of tasks, such as question answering and language inference, without requiring large changes to the architecture for each task [16].

A group of sentences from "s1,s2,s3,..., sn" have two possible outcomes, with xi = {0,1}indicating whether or not a given sentence will be chosen. The output vectors are tokenized instead of sentences thanks to a pre-trained MML (Masked Language Model). Instead of having multiple sentences, it only has two labels, Sa and Sb, and uses embedding to specify the various sentences. These embedding are appropriately changed to produce the necessary summaries (see Fig. 1) [17].

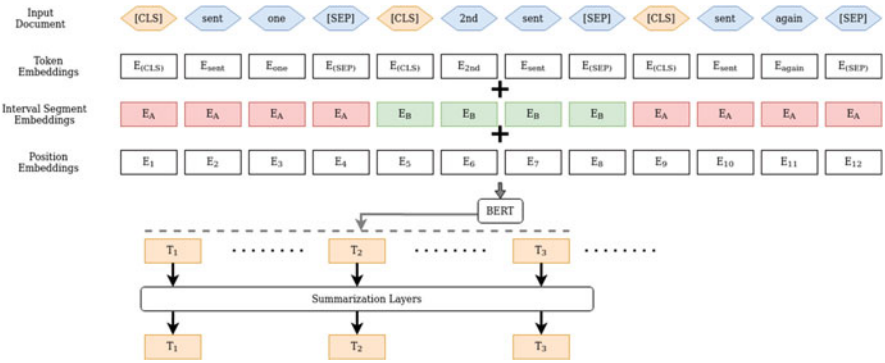

Fig. 1 Process flow of BERT

3.2 Phases in Generating the Summary

3.2.1 Input Document

Sentences from the input source are encoded during this phase. Each sentence has a CLS tag before it and a SEP tag at the end. To group the components of one or more sentences, use the CLS tag.

3.2.2 Interval Segment Embedding

Sentences in the input document will be differentiated during this phase. Each sentence is assigned to one of the labels mentioned above.

Depending on the value of i $\{s_i\}$ might be either$\{E_a$ or $E_b\}$. Essentially, the standard is E_a for even value i and E_b for the odd value i [18].

3.2.3 Embedding

The representation of words in their corresponding vector forms is referred to as embedding. Google makes use of this BERT model feature to help with understanding. Gaining access to various semantic functions, such as those for understanding the purpose of the input document and constructing word-related models.

Our input text is subjected to three different types of embedding before being fed to the BERT model layer:

Embedding tokens: The words are transformed into output vectors with fixed dimensions during this stage. At the beginning and conclusion of input sentences, the CLS and SEP tags are inserted.

3.2.4 Segment Embeddings

Using binary coding, the different inputs are distinguished in this phase.

For example,

Sentence1 – "I love books"

and Sentence2 – "I love sports."

Following the segment embedding operation: [CLS],I,Love,Books,[SEP],I, Love,Sports

Following the segment embedding operation: [0,0,0,0,0,1,1,1], Sentence1 = 0, Sentence2 = 1

3.2.5 Position Embeddings

The BERT model can handle input sequences of up to 512 bytes in length, and its output vector dimension is [512,768]. A word's placement in the input sentence in this phase changes the context of the sentence and should not have the same vectors.

3.2.6 Summarization

The self-attention layer is the primary distinction between recurrent neural networks and Bidirectional Encoder Representations from Transformers (BERT). The model aids in the representation of words and looks for connections between the words.

Simple Classifier: To predict the score Yi, a linear layer and sigmoid function are added to the BERT model. The sigmoid function sets a threshold that determines the range of probability that is mapped to a binary value Y^i [18].

$$Y\hat{\mathrm{i}} = \alpha\left(W_{\mathrm{o}} T_{\mathrm{i}} + b_{\mathrm{o}}\right)$$

3.2.7 Inter Sentence Transformer

The simple classifier is not utilized in this stage. The BERT model is improved by the addition of additional transformer layers, which also help it to better identify the key ideas in the input document [18].

$$h\tilde{}\mathrm{l} = \mathrm{LN}\left(h^{l-1} + \mathrm{MHAtt}\left(h^{l-1}\right)\right.$$

$$hl = \mathrm{LN}\left(h\tilde{}\mathrm{l} + \mathrm{FFN}\left(h\tilde{}l\right)\right)$$

$$h^{0} = \mathrm{PosEmb}(T)$$

where

- T – Sentence output vector by BERT model.
- PosEmb – Positional Embeddings
- LN – Layer Normalization Function
- MHAtt – Multi-head Attention Function
- l – in-depth of the stacked layer.
- Sigmoid output layer – $Y\hat{}\mathrm{i} = \alpha(W_{\mathrm{o}}T_i + b_{\mathrm{o}})$.

4 GPT-2

The GPT tokenizer was used to tokenize the chosen files, although the context size of GPT models limited it to 512 or 1024 tokens (512 or 1024 tokens for GPT and GPT-2, respectively). Identical text files exist for BERT as well; 100 files were selected from the dataset containing the necessary tokens for training (Fig. 2).

Using the tokens that came before it, a language model makes a probabilistic prediction about the token that will come next in the sequence. It gains knowledge of the likelihood that a sentence or string of tokens will appear based on the text examples it has encountered during training. The following conditional probability can be used to represent it:

$$\mathrm{P}\left(w_1^T\right) = \mathrm{P}\left(w_t | w_1^{t-1}\right)$$

where w_{t} is the t^{th} token, and writing sub-sequence $w_i^j = (w_i, w_{i+1}, \ldots, \mathrm{w}_{j-1}, \mathrm{w}_j)$.

The Transformer model variant known as GPT/GPT-2 only has the decoder portion of the Transformer network. They function like conventional uni-directional

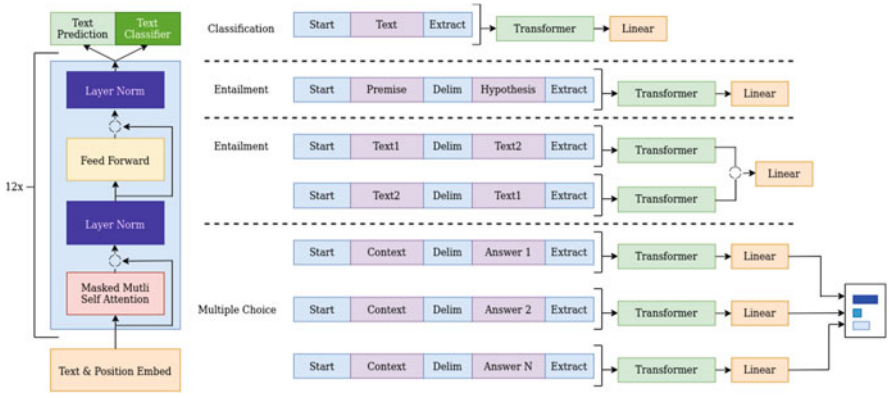

Fig. 2 Process flow of GPT 2

language models thanks to the multi-headed masked self-attention that enables them to focus on just the first I tokens at time step t. These models analyze tokens concurrently, as opposed to sequentially, like RNNs, by simultaneously predicting tokens for all time steps. These role models could be represented by:

$$h_0 = \mathrm{UW_e} + W_\mathrm{p},$$

$$h_i = \text{transformerblock}\,(h_{i-1})\,\forall i \;\epsilon\; [1, n]$$

$$P(u) = \text{softmax}\left(h_\mathrm{n} W_\mathrm{e}^\mathrm{T}\right)$$

where $U = (u_1, u_2,...,u_k)$ are the tokens, n is the number of layers, W_e are the token embedding matrix, and W_p is the position embedding matrix and trained on the language model objective:

$$L(U) = \sum_i^k P\,(u_i \mid u_{i-k}, \ldots\ldots, u_{i-1}; \beta)$$

where $U = (u_1, u_2,...,u_k)$ are the tokens, k is the context size (maximum number of tokens which can be processed at once by model), and P is the conditional probability modeled by our language model with parameters β. Table 1 describes in the test data set with sample title and content [19].

Byte Pair Encoding is used by GPT2 to construct the vocabulary tokens. This implies that the tokens are typically word fragments. GPT-2 can predict the following token in a sequence because it was trained with the intention of causal language modeling (CLM). Using this feature, GPT-2 could produce text that is syntactically coherent. GPT-2 generates fabricated text samples in response to arbitrary input priming the model. The model has the capacity to alter how it looks to correspond with the style and information in the conditioning text [20].

Due to the GPU resource limitation, the abstractive summarization model is a pre-trained distil version of GPT-2. The DistilGPT2 can accept token lengths of up to 1024. It includes 6 transformer decoder layers and 12 attention heads. The Huggingface transformer package's Hugging face GPT-2 is utilized, which is implemented in Pytorch [14].

The GPT-2 performs comparably to supervise baselines on reading comprehension in a no-shot environment. In some activities, like summarizing, it performs the task qualitatively, but based on quantitative measurements, it still performs at a rudimentary level. The zero-shot performance of GPT-2, while suggestive as a research finding, is still a long way from being useful in terms of practical applications [19].

Table 1 Sample test data set

Title	Content
"Today, Macau is paying out again! Each permanent resident will earn 9000 patacas!"	"Macau's 2014 cash-sharing system will go into effect on July 2. MSAR permanent residents and non-permanent residents will get MOP 9000 and MOP 5400, respectively, at that time. The fiscal cost for this plan by the MSAR government is roughly MOP 5659 million. The cash-sharing plan was established in 2008 to distribute the benefits of economic progress to the general populace." [28]
"Cool! Jordan hosts the eighth international special forces tournament, in which Chinese soldiers compete."	"On May 3, local time, the eighth 'warrior race' international special forces competition was conducted in Amman, Jordan, at the King Abdullah Special Warfare Training Center, with 27 teams from various countries competing. China dispatched the armed police snow leopard assault team and the 14-player Hebei Yanshan team. Jordan hosted the seventh 'warrior competition,' an international special forces tournament in which Chinese soldiers participated."

5 Experiment Setup

5.1 About the Dataset

The non-anonymized BCC dataset that Kaggle provided was utilized, which is designed for condensing news items into two to three sentences. The dataset included thousands of text files with an average length of 15 lines and also included a predetermined summary. Using various measures, this summary is used to assess the precision of both models (GPT-2 and BERT). A few more model-specific pre-processing processes were carried out before feeding this data to the models. A few more pre-processing steps specific to the models were performed.

5.2 Training the Models

Some snippets that are relevant in training and summarization are provided here.

```
"from summarizer import Summarizer, TransformerSummarizer

   #this is the code snippet where the summariser is imported and transformer summariser modules after installing transformers from hugging face#

GPT2_model = TransformerSummarizer(transformer_type="GPT2",transformer_model_key="gpt2-medium")

   #this is the code snippet where the parameters are entered for transformer summariser to train the gpt 2 model#

bert_model = Summarizer()

   #this is the code snippet where the summariser is trained for bert model#"
```

5.3 Evaluation Metrics

The term ROUGE, which stands for "Recall-Oriented Understudy of Gisting Evaluation," refers to a set of standards for rating texts that are produced automatically. It is typically employed to assess the effectiveness of a TS algorithm's summary. There are many metrics and ratings that have been proposed in the literature for the evaluation of text summarization results, but ROUGE is the most popular one. Since ROUGE operates similarly on both the abstractive and extractive algorithms, it does not produce outstanding results. In addition, numerous executions are typically preferable to a single one. It ignores the semantic and grammatical precision of the system and human summaries in favor of the overlap of n-grams (represented as a number value).

The process of writing a summary is easier to compare to an abstractive TS task than an extractive one because when writing a summary, a person can try to express his thoughts with new words and phrases after carefully reading and understanding one or more source texts, trying to cover as many topics as possible from the original text [21].

```
“from rouge_score import rouge_scorer
  # a list of the hypothesis documents
hyp = bert_summary
# a list of the references documents
ref = Summary_list
  #  To create aa RougeScore object  for rouge1
scorer = rouge_scorer.RougeScorer(['rouge1'])
  # a results contains precision recall and fmeasure
results = {'precision': [], 'recall': [], 'fmeasure': []}


 # for each of the document  and  hypothesis s pair
for (h, r) in zip(hyp, ref):
    # calculate the ROUGE
    score = scorer.score(h, r)
    # measurements are separated
    precision, recall, fmeasure = score['rouge1']
    #  append them to list in the dictionary
    results['precision'].append(precision)
    results['recall'].append(recall)
    results['fmeasure'].append(fmeasure)  ”
```

5.4 Summary Snippets

Text given as input:
"Safety alert as GM recalls cars.

According to federal officials, the world's largest automaker, General Motors (GM), is recalling roughly 200,000 vehicles in the United States for safety reasons. According to the National Highway Traffic Safety Administration (NHTSA), the greatest recall comprises 155,465 trucks, vans, and SUVs (SUVs). This is due to potential brake system problems. In the year 2004 and 2005 recall of the product affects automobiles, according to GM. The Chevrolet Avalanche, Express, Kodiak, Silverado, and Suburban, as well as the GMC Savana, Sierra, and Yukon, have possible flaws. According to the NHTSA, if the hood was open while driving, pieces from a pressure accumulator in the braking system may shatter, endangering passengers.

It warned that doing so might enable hydraulic fluid to escape, making it more difficult to brake or steer and perhaps resulting in an accident.

Additionally, GM is recalling 19,924 Pontiac Grand Prix cars, SRX SUVs, and Cadillac XLR coupes from the 2004 model year.

This is due to the fact that in severely cold weather the accelerator pedal could not function effectively, necessitating greater braking.

In addition, the automaker is recalling 17,815 2005-model Buick Raniers, Chevrolet Trailblazers, GMC Envoys, and Isuzu Ascenders due to improper windshield installation that could result in a crash.

However, GM emphasized that it was unaware of any injuries connected to the issues.

The recall was made public after GM said last month that it anticipated weaker profitability this year than in 2004.

The largest automaker in the world is struggling with sluggish US sales, losses in its European division, and now a product recall.

In January, General Motors warned that increased healthcare expenditures in North America and decreased earnings at its financial services unit would negatively affect the company's performance in 2005."

Summary given as input
According to GM, the recalled products include cars from the 2004 and 2005 model years.

The NHTSA warned that if the hood was open while driving normally, a pressure accumulator in the braking system might fracture and cause injuries to individuals.

Federal officials have announced that General Motors (GM), the largest automaker in the world, is recalling roughly 200,000 vehicles in the US for safety reasons.

The recall comes after GM said last month that it anticipated weaker profitability this year than in 2004.

In addition, the automaker is recalling 17,815 2005 model-year Buick Raniers, Chevrolet Trailblazers, GMC Envoys, and Isuzu Ascenders because the windshield is improperly installed and could fly out in an accident.

The greatest recall, according to the National Highway Traffic Safety Administration (NHTSA), affects 155,465 pickups, vans, and sports utility vehicles (SUVs).

Summary by BERT model

"Safety alert as GM recalls cars.

According to federal laws, the United States is recalling around 200,000 cars manufactured by General Motors (GM), the world's largest carmaker.

According to the National Highway Traffic Safety Administration (NHTSA), 155,465 trucks, vans, and SUVs have been recalled (SUVs).

In addition, the company is recalling 17,815 Buick Raniers, Chevrolet Trailblazers, GMC Envoys, and Isuzu Ascenders from the 2005 model year because the windshield was mounted incorrectly and might fly out in a collision."

Summary by GPT2 model

"Safety alert as GM recalls cars.

Federal officials have announced that General Motors (GM), the largest automaker in the world, is recalling roughly 200,000 vehicles in the US for safety reasons.

The automaker is also recalling 17,815 2005 model-year Buick Raniers, Chevrolet Trailblazers, GMC Envoys, and Isuzu Ascenders because the windshield is improperly installed and could fly out in an accident.

The recall comes after GM said last month that it anticipated weaker profitability this year than in 2004."

6 Comparison of Results

The numerical values given by ROUGE metrics on precision, recall, and f-measure can be listed and the difference in values given by both the models can be seen in Tables 2 and 3.

Figures 3 and 4 represent the precision, recall, and f-measure given by the ROUGE Metrics. Comparing these bar graphs, it is found that BERT gives better

Table 2 Performance of GPT2

No.	Precision	Recall	F-measure
0	0.437908	0.807229	0.567797
1	0.323529	0.585106	0.416667
2	0.269231	0.555556	0.362694
3	0.270531	0.746667	0.397163
4	0.393162	0.567901	0.464646

Table 3 Performance of BERT

No.	Precision	Recall	F-measure
0	0.483660	0.649123	0.554307
1	0.458824	0.951220	0.619048
2	0.261538	0.548387	0.354167
3	0.415459	0.788991	0.544304
4	0.564103	0.750000	0.643902

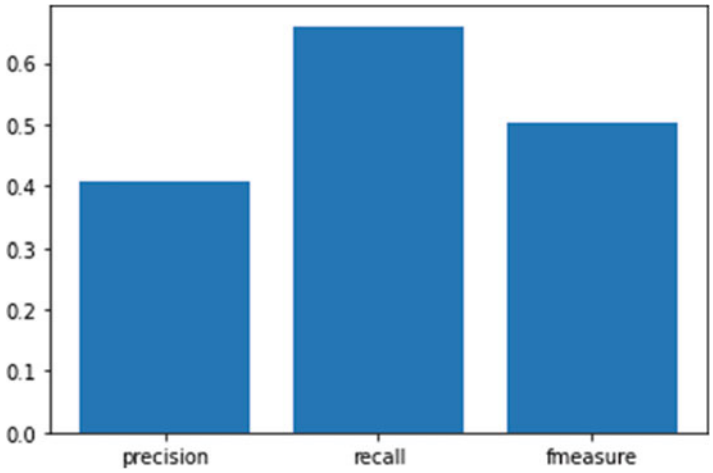

Fig. 3 Performance of GPT2

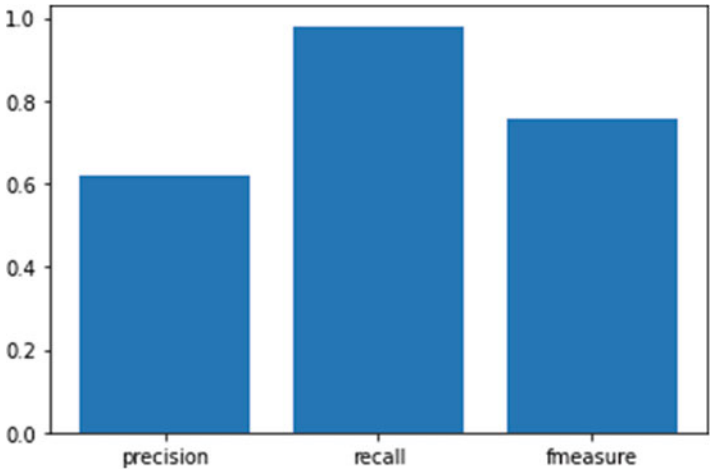

Fig. 4 Performance of BERT

results than GPT 2. These bar graphs are just representations of two random summaries that were generated by the respective Transformer models.

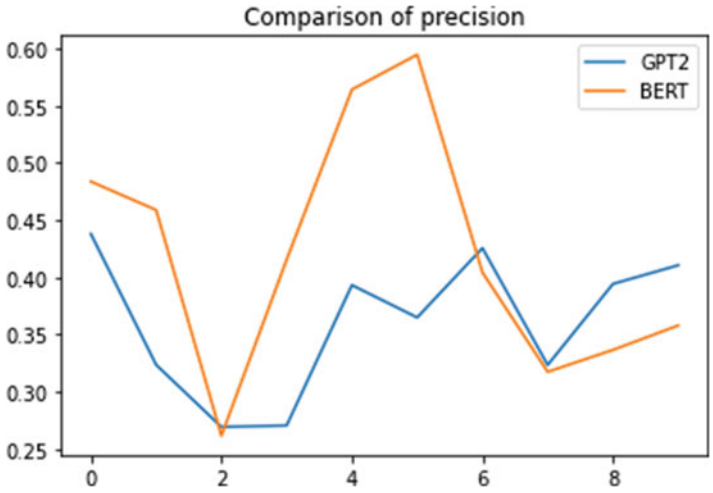

Fig. 5 Precision comparison of GPT2 and BERT

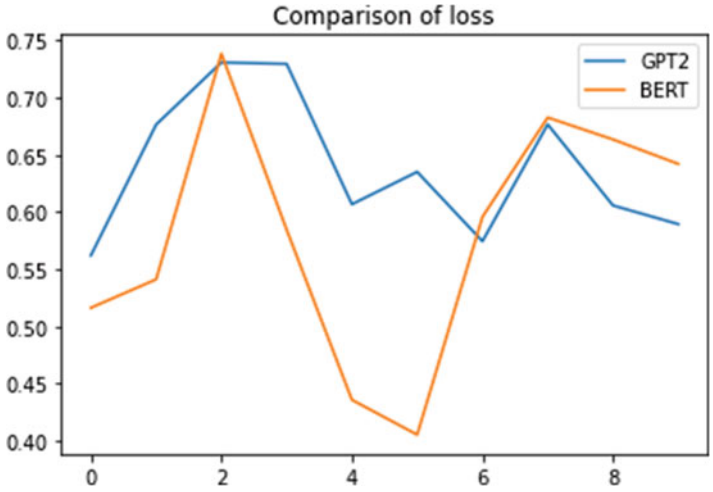

Fig. 6 Loss comparison of GPT2 and BERT

Figure 5 can be used to visualize the trend in which both the models perform in terms of precision. Here it can be seen that the precision given by GPT2 is lower compared to the one given by BERT.

From Fig. 6, the loss function of the BERT model seems to be comparatively better than GPT 2.

7 Conclusion

From all the visualizations and analysis done using ROUGE metrics on BERT and GPT 2 models, it is concluded that BERT gives better results than GPT 2. This conclusion has been reached by analyzing the numerical values obtained by ROUGE metrics and the results that the graph shows. BERT gives an average precision of 0.40 whereas GPT2 gives 0.38. Comparing this and the other factors mentioned by the metrics, it is concluded the BERT model for Big Data Analytics using summarization.

References

1. Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2022). T-BERTSum: Topic-aware text summarization based on BERT. *IEEE Transactions on Computational Social Systems, 9*(3), 879–890. https://doi.org/10.1109/TCSS.2021.3088506
2. Babar, S., Tech-Cse, M., & Rit (2013). Text summarization: An overview.
3. Gupta, A., Chugh, D., & Katarya, R. (2022). Automated news summarization using transformers. In *Sustainable advanced computing* (pp. 249–259). Springer.
4. Suleiman, D., & Awajan, A. (2020). Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges. *Mathematical Problems in Engineering, 2020*.
5. Shini, R. S., & Kumar, V. A. (2021). Recurrent neural network based text summarization techniques by word sequence generation. In *2021 6th international conference on inventive computation technologies (ICICT)* (pp. 1224–1229). IEEE.
6. Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science, 37*(4), 405–417.
7. Mahajani, A., Pandya, V., Maria, I., & Sharma, D. (2019). A comprehensive survey on extractive and abstractive techniques for text summarization. In *Ambient communications and computer systems* (pp. 339–351).
8. Liu, Y., & Lapata, M. (2019, August 22). *Text summarization with pretrained encoders*. arXiv preprint arXiv:1908.08345.
9. Rahman, M. M., & Siddiqui, F. H. (2019). An optimized abstractive text summarization model using peephole convolutional LSTM. *Symmetry, 11*(10), 1290.
10. Vig, J. (2019, June 12). *A multiscale visualization of attention in the transformer model*. arXiv preprint arXiv:1906.05714.
11. Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications, 78*(1), 857–875.
12. Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 93–98).
13. Nagalavi, D., Hanumanthappa, M., & Ravikumar, K. (2019). An improved attention layer assisted recurrent convolutional neural network model for abstractive text summarization. *INFOCOMP Journal of Computer Science, 18*(2), 36–47.
14. Kieuvongngam, V., Tan, B., & Niu, Y. (2020, June 3). *Automatic text summarization of covid-19 medical research articles using bert and gpt-2*. arXiv preprint arXiv:2006.01997.
15. Miller, D. (2019, June 7). *Leveraging BERT for extractive text summarization on lectures*. arXiv preprint arXiv:1906.04165.

16. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. Conf. North Amer.- Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol* (Vol. 1, pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1423
17. Abdel-Salam, S., & Rafea, A. (2022). Performance study on extractive text summarization using BERT models. *Information, 13*(2), 67.
18. Liu, Y. (2019, March 25). *Fine-tune BERT for extractive summarization*. arXiv preprint arXiv:1903.10318.
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.
20. Montesinos, D. M. (2020, September 10). *Modern methods for text generation*. arXiv preprint arXiv:2009.04968.
21. Barbella, M., & Tortora, G. *Rouge metric evaluation for text summarization techniques*. Available at SSRN 4120317.
22. Bharathi Mohan, G., & Prasanna Kumar, R. Survey of text document summarization based on ensemble topic vector clustering model. In P. P. Joby, V. E. Balas, & R. Palanisamy (Eds.), *IoT based control networks and intelligent systems* (Lecture notes in networks and systems) (Vol. 528). Springer. https://doi.org/10.1007/978-981-19-5845-8_60
23. Mohan, G. B., & Kumar, R. P. (2022). A comprehensive survey on topic modeling in text summarization. In D. K. Sharma, S. L. Peng, R. Sharma, & D. A. Zaitsev (Eds.), *Micro-electronics and telecommunication engineering . ICMETE 2021* (Lecture notes in networks and systems) (Vol. 373). Springer. https://doi.org/10.1007/978-981-16-8721-1_22
24. Kalpana, G., Kumar, R. P., & Ravi, T. (2010). Classifier based duplicate record elimination for query results from web databases. In *Trendz in Information Sciences & Computing (TISC2010)* (pp. 50–53). https://doi.org/10.1109/TISC.2010.5714607
25. Assegie, T. A., Rangarajan, P. K., Kumar, N. K., & Vigneswari, D. (2022). An empirical study on machine learning algorithms for heart disease prediction. *IAES International Journal of Artificial Intelligence (IJ-AI), 11*(3), 1066. 10.11591/ijai.v11.i3.pp1066-1073.
26. Mohan, G. B., & Kumar, R. P. (2022). Lattice abstraction-based content summarization using baseline abstractive lexical chaining progress. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-022-01080-y
27. Yang, Z., Dong, Y., Deng, J., Sha, B., & Xu, T. (2021). Research on automatic news text summarization technology based on GPT2 model. In *2021 3rd international conference on artificial intelligence and advanced manufacture*. https://doi.org/10.1145/3495018.3495091