Text Summarization using BERT and T5

Anjali Pal
anjali.pal.21@ucl.ac.uk

Kanupriya
kanupriya.21@ucl.ac.uk

Langlang Fan
langlang.fan.21@ucl.ac.uk

Vanessa Igodifo
vanessa.igodifo.21@ucl.ac.uk

Abstract

Transformer models have grown in popularity in recent years. This is primarily due to the self-attention mechanism in their architecture that gives differential weights to the significant portion of the text. This feature also allows parallelization compared to traditional methods, thereby reducing training time. Although all transformer based approaches are really good, there is still a challenge to decide which transformer model will perform better with a new dataset. In this paper, we propose a few experiments for text summarization using the WikiHow dataset. We use two models in a contest to outperform the other- BERT-large (extractive text summarizer) and T5-small (abstractive text summarizer). We conducted experiments on various text lengths and compared them using ROUGE scores. Following this, we conducted an experiment to determine which of the two models would yield better results in terms of information retrieval.

1 Introduction

The Modern era is ruled by data and language which is the nucleus of human communication. Many great minds have worked and have been working on language modeling and sequence patterns which led to the inference that machines can learn.

As a result, machines have gradually learnt to predict probable sequence of words. In today's world our lives are highly dependent on natural language processing functions including but not limited to social media, e-mail, language translations and web search engines.

Neural Networks are considered the backbone of Deep Learning and NLP is highly dependent on them for extracting and processing complex information from various data. The three dominant neural networks are-

1. Recurrent Neural Networks (RNNs) which are derived from feedforward neural networks and are designed to interpret sequential information.

2. Convolutional Neural Networks (CNNs) which are most commonly applied in image recognition and processing that is specially designed to analyze visual data.

3. Long Short Term Memory (LSTMs) which are based on RNN architecture but in contrast to standard feedforward neural networks, LSTM has feedback connections capable of learning long order dependency in sequence prediction problems.

However, the use of these networks entails a huge cost in terms of computation and machine power. In December 2017, Google Brain members and Google Research published an article (Vaswani et al., 2017) and we were introduced to Transformers. It was both revolutionary and disruptive and gave us an alternative for RNNs and CNNs.

Transformers are industrialized, homogenized post-deep learning models designed for parallel computing on supercomputers. Through homogenization, one transformer model can carry out a wide range of tasks with no fine-tuning. Transformer encoders and decoders contain attention heads that train separately on billions of records of raw unlabeled data, and can run on separate GPUs which opens the door to billion-parameter models (Rothman, 2021). Thus, they rapidly became vitally important in the field of Natural Language Processing as the state-of-the-art transformers models outperformed the existing NLP models by training more quickly than the former architectures and have produced better evaluation outcomes.

As data continues to grow, automated text synthesis has become an integral component to compress vast amount of information. The main idea behind it is to understand the major theme of the document or article and condense them into a shorter summary under a length limit which would contain the most important points from the text. In our work, we create short summaries of our articles which are at most one third of the article length. There are two summarization methods to do this - Extractive and Abstractive. Both of them enable quick use of relevant information in these documents, thereby reducing both the cost and time computation of our problem.

In this work, we use dataset from Wikihow knowledge base (Koupaee and Wang, 2018) which consists of how-to-articles edited by the readers. We deploy bidirectional encoder representations from transformers (BERT) and text-to-text transfer transformer (T5) models and concentrate on summarizing these articles from the dataset. The Rogue scores are then used to assess the performance of our models and compare them. Abstractive methods offer a novel alternative by constructing a semantic representation of the text leading to new words paraphrasing the article while the extractive methods identify important sections of the document and produce summaries using subsets of the original article. These summaries can therefore be used at a later stage for information retrieval which we demonstrate using the BM25 model.

2 Literature Review

Understanding "language" has historically been a difficult task for computers (Haugeland, 1979) which led to making language models for specific tasks. Training these models consumed a lot of time as everyone had to make their own language model before training it for use-case. Therefore, there was a need to create a learner which could be pre-trained from a related source and directly be used for any domain. This was the motivation behind transfer learning. (Pan and Yang, 2010)

With the help of pre-training language models, excellent results were obtained in downstream natural language processing tasks (Dai and Le, 2015) (Radford et al., 2018). Google AI team revolutionized NLP in 2018 with BERT- bidirectional encoder representations from transformers (Devlin et al., 2018). It could solve multiple tasks including sentiment analysis, semantic role labeling, sentence classification and the disambiguation of polysemous words. This recent success of transfer learning ignited by GPT, ULMFiT, ELMo, and BERT led to development of a huge diversity of new methods like XLNet, RoBERTa, ALBERT, Reformer, and MT-DNN (Roberts and Raffel, 2020). Following this, Google AI came up with another transformer which was based on encoder-decoder architecture known as T5 (Raffel et al., 2020). This architecture involved converting every problem to text-to-text task and then training and fine tuning it in a supervised/unsupervised way. Both BERT (Miller, 2019) and T5 (Garg et al., 2021) perform well on Natural Language Processing tasks though no research is available for comparison between the two models for text summarization. Thus, we use these two models - BERT and T5 to generate text summaries and compare their performance on the WikiHow dataset (Koupaee and Wang, 2018) and conduct experiments to determine whether the models differ significantly under different conditions.

3 Methods

3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based NLP technique developed and pre-trained by Google. It overcomes the limitations of Recurrent neural networks and other long term dependency networks. It uses a robust flat architecture with inter-sentence transform layers in order to achieve best results in summarization (Vashisht).

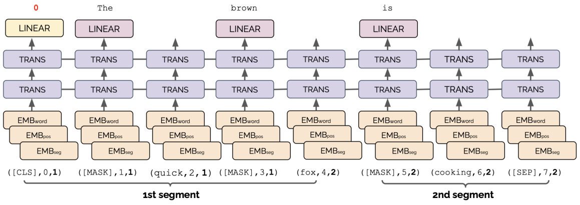

Figure 1: BERT Architecture for summarization (Chalkidis, 2022)

For a list of sentences, we have two possibilities, either it will be selected or not. Since the model is trained as a masked model, the output vectors are tokenised. There are 3 kinds of embedding-

1. Word embedding where words are converted to fixed dimension vectors and each sentence is preceded by [CLS] and succeeded by [SEP].

2. Segment embedding differentiate inputs using binary coding.

3. Position embedding are used to retain contextual text information.

The sum of these three embedding is the input of the TRANS (Transformer) layer which is a combination of encoder and decoder layers. Each encoder incorporates self-attention with a feedforward network with the decoder also having a similar architecture with another layer of attention between self-attention and feedforward network which helps to keep the focus and emphasis on important words.

In our work, We use the BERT-Large model with 24 transformer layers, 1024 hidden layers, 16 attention heads and 336 million parameters and the maximum length of the summary is set to be one-third of the actual length of the text.

3.2 T5

Text-To-Text Transfer Transformer (T5) was developed and pre-trained by Google in 2020. Google AI also developed the Colossal Clean Crawled Corpus (C4) - a pre-training dataset. T5 was then pre-trained on C4 for denoising, and corrupting span objective with an Encoder-Decoder architecture. It is then fine-tuned for a downstream task by means of a supervised objective. The architecture of T5 is pretty similar to BERT.

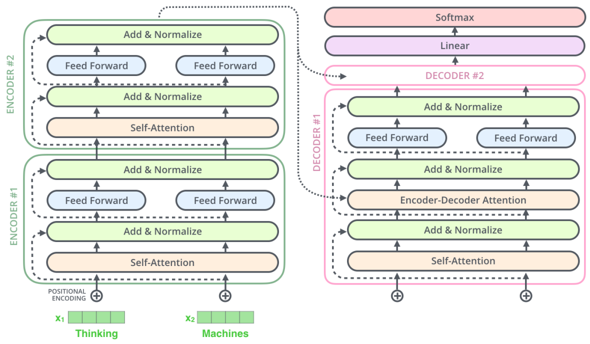

Figure 2: T5 Architecture for summarization (Alammar, 2018)

The encoder is composed of blocks with a self-attention layer and feedforward network whose inputs have been normalized using layer normalization (Ba et al., 2016). A simpler version of layer normalization is used in which activation is simply re-scaled and calculated with no additive bias. Following this, a residual skip connection (He et al., 2016) is introduced to map the input to the output. Dropout is applied in feedforward network, skip connection, attention weights and input/output of the whole stack. The decoder is almost similar with the addition of attention layer. The self-attention in the decoder makes it possible for the model to utilize past outputs. The final output of the decoder is passed through a dense softmax output. Thus, all the tasks are in text-to-text format, unlike BERT, where the inputs are strings of text but the outputs are either a class label or a span of inputs. This framework enables us to use the same model for different tasks.

In our work, the maximum length of the summary is set to be one-third of the actual length of the text. Due to computational restrictions, we use the T5 small. It has 6 blocks (each block comprising of self-attention, encoder-decoder attention, and a feedforward network). The feedforward network in each block has a dense layer with 2048 output dimensions, followed by ReLU and another dense layer. The “key” and “value” matrices of all attention mechanisms have an inner dimensionality of 64 and all attention mechanism have 8 heads. All other sub-layers and embeddings have dimensionality of 512. Therefore, the model has around 60 million parameters. The early-stopping parameter has been set as "True".

4 Experiments

Dataset - WikiHow dataset (Koupaee and Wang, 2018) contains 215,365 articles obtained from WikiHow data dump. These articles cover a wide range of topics and writing styles. Each article has several paragraphs and each paragraph has a headline that summarizes it.

In our research, we are excluding articles whose word length is less than 30 because the summaries of such short articles are more likely to be flawed or almost identical to the text. Therefore, it is assumed that articles of such few words do not require summarization.

Due to BERT and T5 small’s inability to process articles of length longer than 512 words without automatic truncation (Devlin et al., 2018), we removed articles and text whose number of words exceeded that. Following this, stratified sampling is performed to reduce the size of the data set. This is done due to the memory constraints. In an attempt to approximately retain the distribution of the original data, we have also removed articles whose word length only occurs once in the entire corpus for simplification as this allows us to sample the dataset in a stratified manner. After sampling, we have 70,473 articles of varied lengths, where the maximum length does not exceed 512 tokens. The sample distribution of the data is shown in the Figure 3.

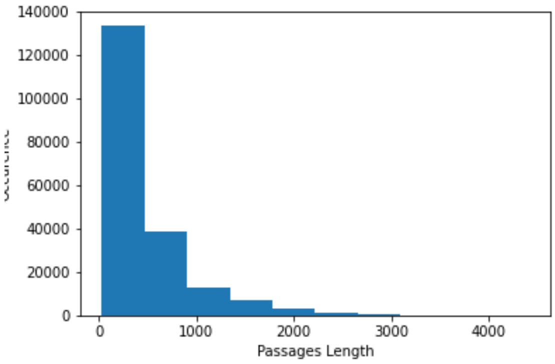

Figure 3: Sample distribution of the processed data with passage length on X-axis and number of passages on Y-axis

Testing Metrics - We use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) as standardized metric for testing our summaries from BERT and T5. These metrics compare the automatically generated summaries to some references. Since the real data is an explanation of the headlines, we assume that the headlines are the best representation of the entire article. Thus, in our experiments, we set this reference as the headlines of the articles in the dataset.

More specifically, we use ROUGE-1 (based on overlap of uni-grams) (Lin and Hovy, 2003) , ROUGE-2 (based on overlap of bi-grams) (Lin and Hovy, 2003) , ROUGE-L (based on overlap of longest common sub-sequence on sentence level) (Lin and Och, 2004) and ROUGE-L-SUM (based on overlap of longest common sub-sequence on summary level) (Lin and Och, 2004).

For all the ROUGE metrics, we use the ROUGE F1 score, as it gives us a measure of model performance that not only relies on capturing maximum words (recall) but also on reducing irrelevant words (precision).

4.1 Compare BERT and T5

In this experiment, ROUGE scores are computed for the summaries created by each model and compare those scores. This helps us evaluate which model works best on the WikiHow dataset (for articles with a length of less than 512, due to the memory constraints) . For the ROUGE score, we used headlines of the articles as references as indicated above. The model with a higher ROUGE score performs better.

Metric	BERT	T5
ROUGE-1	0.35	0.22
ROUGE-2	0.06	0.05
ROUGE-L	0.22	0.16
ROUGE-L-SUM	0.25	0.18

Table 1: Table for ROUGE metrics of BERT and T5

ROUGE-1 scores are significantly higher than ROUGE-2, which is because uni-grams are more common to be found in the summary and reference, rather than bi-grams.ROUGE-L searches for the longest common subsequence which appears in the same relative order and is computed on individual sentences. Thus, the one with a higher ROUGE-L will have a more similar sentence structure with reference summaries. ROUGE-L-SUM does the same for the entire summary as opposed to each sentence.
Thus, we infer that the BERT model gives better results and outperforms the T5-small model for all the ROUGE metrics.

4.2 Find a relationship between short text length and ROUGE score

In this experiment, we examined articles with text lengths below 512 words. Our goal is to understand whether there is a linear relationship between the length of the original text and the metrics. A positive correlation would indicate that a longer text would provide a better summary where as a negative correlation would suggest that reducing the length of the text is preferable for the summary.

Metric - BERT	Correlation
ROUGE-1	0.02
ROUGE-2	0.03
ROUGE-L	0.01
ROUGE-L-SUM	0.02

Table 2: Table for correlation between article length and ROUGE metric for BERT

From Table 2, we infer that there is practically no correlation between short text lengths and ROUGE score in BERT. This means that, if the length of the article length is less than 512 words, then regardless of the length of the article, there is no impact on the ROUGE score.

From Table 3, we deduce that ROUGE-1 and ROUGE-L-SUM have a weak positive correlation

Metric - T5	Correlation
ROUGE-1	0.21
ROUGE-2	0.08
ROUGE-L	0.13
ROUGE-L-SUM	0.18

Table 3: Table for correlation between article length and ROUGE metric for T5

with the length of the articles. This indicates that the greater the length of the article, the greater the ROUGE-1 and ROUGE-L-SUM.

4.3 Compare ROUGE score for large text lengths

In this experiment, we examine the articles whose text length is between 30 words to 2048 words. Since, we have been unable to find any strong relationship for short text lengths, we now aim to infer if there exists a linear relationship between the length of the original text and the metrics taking into account all text lengths.

Metric - BERT	Correlation
ROUGE-1	0.26
ROUGE-2	0.16
ROUGE-L	0.05
ROUGE-L-SUM	0.18

Table 4: Table for correlation between article length and ROUGE metric for BERT

From Table 4, we deduce that, BERT's ROUGE-1, ROUGE-2 and ROUGE-L-SUM show a weak positive correlation with the length of the article. This would mean that as the length of the article increases, the above ROUGE metrics might increase. Moreover, ROUGE-L has almost no correlation to the length of the article.

Metric - T5	Correlation
ROUGE-1	0.23
ROUGE-2	0.05
ROUGE-L	0.16
ROUGE-L-SUM	0.18

Table 5: Table for correlation between article length and ROUGE metric for T5

From Table 5, we conclude that ROUGE-1 still shows a weak positive correlation with length of the text. This is followed by ROUGE-L-SUM and ROUGE-L and ROUGE-2 almost has no correlation with the length of the text.

4.4 Compare summarization from T5 and BERT for information retrieval using BM25

Summaries are useful for information retrieval. Instead of parsing the original article, if they parse through their representative summaries, similar results could be generated, with a lot less computational power. BM25 is a popular and effective ranking algorithm that relies on a binary independence model. Given a query, it is used to estimate the relevance of a document. In this experiment, document refers to summaries by BERT and T5, and query means the title of the article. Therefore, we have 2 documents for each query and we aim to find which one of the two is more relevant. The higher the BM25 score is, the greater the relevance of the summary to the query. The BM25 score is given by:

$$\sum_{i\epsilon Q} log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \frac{(k_1 + 1)f_i}{K + f_i} \frac{(k_2 + 1)qf_i}{k_2 + qf_i}$$

Figure 4: Formula for BM25

In the above formula, r and R are zero as we don't have any prior relevance information. f_i and qf_i are the number of occurrences of the term in the document and the query, respectively. k_1, k_2 and K are parameters whose values are set empirically. K is given by:

$$K = k_1((1 - b) + b \frac{document-length}{avg-document-length}$$

Figure 5: Formula for K in BM25

We set $k_1 = 1.2$, $k_2 = 100$ and b = 0.75.
After inserting these values into the formula, we obtain two scores for each title in our data (one for BERT and the other for T5). Then, we find out which of the two models will work best, if summaries are used for retrieval instead of the original text.

Model	Relevance Percentage
BERT wins	44%
T5 wins	32%
Tie	24%

Table 6: Table for Relevance Percentage

From Table 6, we can infer that BERT wins 44% of the time while T5 wins 32% of the time. Also, the two models tie 24% of the time. This might mean that BERT-large performs better than T5 mostly but not always.

5 Results

In Experiment 1, we deduce that BERT performs better in all the ROUGE metrics. The reason behind this is that we compare BERT-Large and T5 Small. BERT-Large has 336M parameters whereas T5 has 60M parameters. Due to this five-fold difference BERT performs much better than T5 but it does so at the cost of a higher training time and memory. Moreover, T5 gives an abstractive summary and ROUGE metrics do not take into account the semantic meaning of words and simply measure syntactical matches. This could mean a lower ROUGE metric score for T5, despite of a better summary than BERT.

In Experiment 2, we only compute scores using articles whose length is less than 512. Here, we infer that BERT's ROUGE metrics show no correlation with the text length of the articles while T5's ROUGE-1, ROUGE-L and ROUGE-L-SUM metric show a weak positive correlation with text lengths of the articles. This could be due to the fact that, for BERT, we use an extractive summarization and so the model is capable of recognizing important sections more easily. As a result, the length of the article has no impact on the ROUGE metrics. Whereas for T5, since this is an abstractive summarizer, the metrics might get better with longer length of the articles.

In Experiment 3, all the articles between the length of 30 and 2048 (which is the maximum possible length for the two models) were used to compute the correlations. It can be inferred that BERT's ROUGE-1,ROUGE-2 and ROUGE-L-SUM have a weak positive correlation with the length of the article. The reason for this might be that as the length of the article increases, BERT has to summarize more and that's why it extracts more uni-grams and bi-grams from the original text. The T5 correlations are similar to Experiment 2, signifying that ROUGE metrics for T5 have a weak linear relationship with the length of the articles. Furthermore, the ROUGE-L-SUM correlation for BERT and T5 is equally correlated with the length of the articles. This indicates that the overall score for the summary of the two models depends on the length of the article.

In Experiment 4, we use the summaries of the two models as documents and title of the article as a query for information retrieval. Following that, we use the BM25 ranking algorithm to rank the summaries and see which model performs better. The two models work and perform the same way for 24% of the cases, which means they are both good at summarizing but we found BERT to be more successful in most cases. Since we use T5-Small, which is almost one-fifth of the BERT-large model, T5 is more efficient than BERT in 32% of the cases, implying that T5 fact a rather strong competitor that not only runs faster but also uses less memory.

6 Conclusion and Discussion of Future Work

The results we have achieved may not be comparable with the current state-of-art research. Nevertheless, they are satisfactory in view of the various constraints:

1. Due to the limited time frame and computational resources, we worked with T5-small instead of T5-base which had more parameters, and could therefore have yielded better results.

2. We could not use the entire dataset because it was too large. Thus, in order to reduce the size of the dataset, a stratified sample was taken which reduced the dataset to one-fourth of its original size.

There is a great deal of future work possible in this area. First, we can implement batch training for passages greater than 2048 in length. This will help us identify and understand the differences in model performance for extremely important data sequences. Second, the dataset did not contain any topic-wise information. This information could be added to the data through annotations that can further be used to verify if a model performs better on a specific topic. This is due to the fact that different topics have a different level of abstractedness which makes it difficult for models to summarize it and would therefore affect the quality of the summary. Since, in our work, we used T5 as abstractive and BERT as extractive, we would probably obtain different results for different topics. In addition to this, our evaluation metric also needs to be modified because ROUGE metrics does not take into consideration the semantic meaning of the word. This means that T5 (abstractive model) has a disadvantage as compared to BERT (extractive model) when we use this metric. Therefore, changes are needed in the ROUGE metric to take into account the semantic meaning of words.

Although, T5 is a great competitor in-spite of having one-fifth of the parameters as compared to BERT, we see that BERT-Large performs better and is more powerful than T5-Small for the WikiHow dataset. This could mean that if we use T5-base which has more parameters, T5-base could perform better than BERT-large.

7 Appendix

Training Time: We use NVIDIA Tesla K80 GPU provided by Google Colaboratory to train our model. Given this hardware, it takes 40 minutes to train for one epoch.
Github Repository: The repository that contains all code, results and dataset used can be found at Repository Link
Video: The video explaining the whole project can be found at Video Link.

Acknowledgement

This research is produced as a part of coursework for Statistical Natural Language Processing (COMP0087). We would like to thank Prof. Sebastian Riedel, Prof. Tim Rocktäschel and Prof. Pasquale Minervini for their support. We would also like to express our gratitude to Robert Kirk for providing us guidance throughout the project and reviewing our work.

References

Jay Alammar. 2018. The illustrated transformer.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Ilias Chalkidis. 2022. A nlp story from bag of words to muppet show.

Andrew M Dai and Quoc V Le. 2015. *Semi-supervised sequence learning*. Advances in neural information processing systems.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. 2021. News article summarization with pretrained transformer.

John Haugeland. 1979. *Understanding Natural Language*. Journal of Philosophy, Inc.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics.

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures.

Sinno Jialin Pan and Qiang Yang. 2010. *A Survey on Transfer Learning*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Adam Roberts and Colin Raffel. 2020. Exploring transfer learning with t5: the text-to-text transfer transformer.

Denis Rothman. 2021. *Transformers for Natural Language Processing*, volume 1. Packt, Livery Place, 35 Livery Street, Birmingham B3 2PB, UK.

Ashutosh Vashisht. Bert for text summarization.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.