Investigating effectiveness of extractive text summarization using transformer-based embeddings and k-means clustering

Xi Zhang
University of Toronto
Toronto, ON
xix.zhang@mail.utoronto.ca

Emre Cargin
University of Toronto
Toronto, ON
emre.cagin@mail.utoronto.ca

Abstract

Transformer-based models have become increasingly popular and the state-of-the-art architecture due to their robustness, leading to a surge in major breakthroughs in downstream NLP tasks. Among the various NLP tasks, text summarization has become important due to its practicality in everyday life. In this paper, we aim to evaluate the effectiveness of the model proposed by Miller (2019) in [3], which combines BERT and k-means clustering to perform extractive text summarization. Additionally, we extend [3] by evaluating the performance of combining GPT-2 and k-means clustering on two popular text summarization datasets. Based on our experimental results, we will compare the two models, discuss their limitations, and provide insights into future improvements.

1 introduction

In the era of information overload, the ability to extract key information from large volumes of text has become very beneficial, which makes the NLP task of text summarization crucial. There are mainly two types of text summarization, namely extractive and abstraction summarization. Extractive summarization, which is what our research focuses on, is the method of naively selecting a subset of sentences from the original text as the summary. On the other hand, abstractive summarization generates new text based on information in the original text and tries to convey the same content in a more concise way, which is by nature more challenging. In [3], the author proposed an intuitive extractive text summarization model by using k-means clustering on sentence embeddings retrieved from BERT. However, there is a lack of objective evaluation in [3], which makes the model's effectiveness unknown to readers. In this paper, we extend [3] by investigating the effectiveness of combining two Transformer-based embedding architectures, namely BERT and GPT-2, with k-means clustering, in the hope of generating concise and accurate text summaries. Our study evaluates the two models on the CNN-DailyMail dataset and the Reddit TIFU dataset using the ROUGE F1 metrics.

2 Background

Please check Optional Appendix for background on embedding mechanisms and k-means clustering.

3 Model

In this section, we will introduce the model proposed in [3]. As previously mentioned, the model first uses BERT for sentence embeddings, then feeds the sentence embeddings into the k-means clustering algorithm. In our experiments, we tested another model not mentioned in the paper which

is to combine GPT-2 with k-means clustering. However, due to the similar natures of the two models, we will just introduce the architecture of the model in [3], as the only difference between the two models is the change in the embedding architecture. The architecture of the proposed model in [3] is shown in Figure 1.

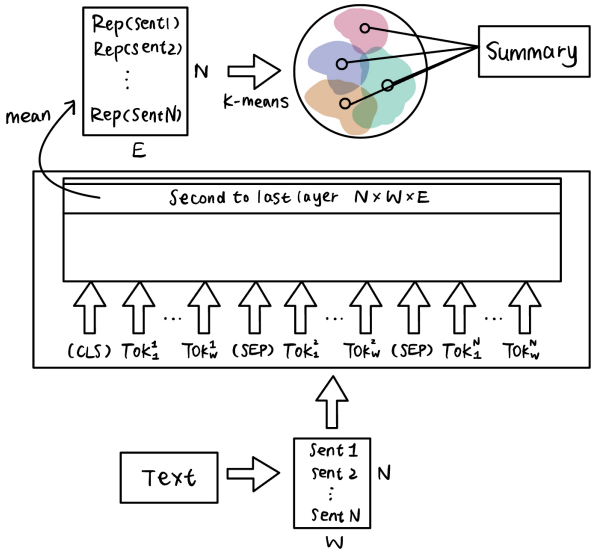
3.1 Main Idea

The main idea of the model in [3] is that the embedding outputs from BERT serve as a representation of the input sentences in terms of their syntactic and semantic properties. Therefore, by clustering the sentences in a text corpus according to their embeddings, we would be able to summarize the original text by selecting a subset from each cluster as the most representative sentences.

3.2 Sentence Representation from BERT

Let N be the number of sentences and E be the embedding dimensions. The goal is to obtain the best representations of the sentences, that is, the $N \times E$ matrix for clustering. A natural thing to do is to select the [cls] layer. However, Miller (2019) suggests that it is actually the second to last layer that produced the best embeddings for representations, since the final layer is biased towards classification tasks due to the pre-training of BERT [3]. Since the output of the second last layer is $N \times W \times E$ where W is the number of tokenized words, we average the embeddings of the tokens for each sentence to obtain the $N \times E$ sentence embeddings. Finally, the sentence embeddings would be fed into the k-means clustering algorithm and the sentences closest to each cluster centroid would be selected as part of the final summary.

Figure 1: Figure 1: Model architecture



The source code for the model proposed in [3] was made accessible to the public by the author Miller (2019) on Github [8], and the models in [3] were also made available on PyPI [9].

4 Experiments

4.1 Data

In order to have a more unbiased evaluation on the model performance, we tested the models on two datasets of very different natures, namely the CNN-Daily Mail dataset [4, 10] and the Reddit TIFU dataset [5, 11].

4.1.1 CNN-Daily Mail Dataset

This dataset consists of formal news articles and their corresponding summaries obtained from CNN and Daily Mail news websites. The articles are written in a professional tone. This dataset consists of 286,817 training pairs and 11,487 test pairs [10].

4.1.2 Reddit TIFU Dataset

This dataset consists of Reddit posts and their corresponding summaries from the TIFU subreddit. This dataset contains text that is casual and conversational due to its nature. This dataset consists of 33,711 training pairs and 4,214 test pairs [11]. It is worth mentioning that both the original text and the summaries are on average shorter than those of the CNN-Daily Mail dataset.

4.2 Preprocessing

In the CNN-Daily Mail dataset, the reference summaries have newline characters in between sentences as well as a lot of unwanted spaces. In order to ensure consistency between the reference and generated summaries and optimality during testing, we processed the data to remove unwanted characters. In the Reddit TIFU dataset, we decoded the byte strings so that we could proceed with the experiments.

4.3 ROUGE Metrics

To compare the performances of our two models, we used the ROUGE score on the test set. ROUGE score evaluates the similarity between the summary generated by the model with reference summaries that serve as the ground truth. In this paper, we used the ROUGE-N score and the ROUGE-L score. ROUGE-N score measures the overlap of N-grams between the model-generated summary and the reference summary, while ROUGE-L score measures the longest common subsequence (LCS) between the model-generated summary and the reference summary, where LCS is the longest sequence of words not necessarily consecutive that are common to the model-generated summary and the reference summary [7]. More specifically, in our experiments, we chose the F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L as the final metrics.

4.4 Experimental Results

We ran the two models on the test split for each dataset, and took the mean of the ROUGE F1 scores calculated using the generated summaries and the reference summaries across all test samples. The performance of the two models on the two datasets can be seen in Table 1 and Table 2.

Table 1: Performance of BERT-large-uncased + k-means clustering

Dataset	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
CNN/Daily Mail	30.16	11.55	18.83
Reddit TIFU	15.45	2.81	10.59

Table 2: Performance of GTP-2 + k-means clustering

Dataset	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
CNN/Daily Mail	30.59	11.78	19.07
Reddit TIFU	15.74	2.74	10.83

5 Summary and Discussion

In this paper, we extended [3] by investigating the effectiveness of combining two Transformer-based embedding architectures, namely BERT and GPT-2, with k-means clustering. We evaluated the performance of the two models on two popular datasets, namely the CNN-Daily Mail dataset and the Reddit TIFU dataset.

Based on the experimental results, the model using GPT-2 embeddings marginally outperforms

the model using BERT embeddings on all metrics across the two datasets but ROUGE2 for Reddit TIFU. However, compared to the benchmarks, both models underperform by roughly 10% on the CNN-Daily Mail dataset and 15% on the Reddit TIFU dataset. We believe there are two aspects that could be improved on. First, the current approach to obtaining sentence embeddings is to naively average the embeddings for individual tokens, which might not be the most robust way to represent a sentence. Second, Euclidean distance might not be the optimal metric when determining sentence similarity, and therefore k-means clustering might not be the optimal solution to picking the most representative sentences in a corpus.

One major observation is that the models perform significantly worse on the Reddit TIFU dataset. As mentioned in [3], the models do not seem to perform well on words that need further context such as "this" and "that", as it is difficult to determine the true context. Since Reddit TIFU is a more casual and conversational dataset, it is expected to contain more demonstrative pronouns in its samples, which might explain the poor performance.

We believe there are two ways to improve the rigor of the experiments. First, when measuring the performance of the models on the CNN/Daily Mail dataset, we took into account factors like capitalization and punctuation marks, which potentially introduced noise in the results. The performance could potentially be improved with better data preprocessing. Second, the length of the generated summaries might not be ideal for optimal performance. In the models, the user gets to choose the number of sentences in the generated summaries by controlling the "ratio" parameter which controls the ratio between the generated summary and the original text. In our experiments, the default ratio of 0.1 was used. However, carefully tuning the number of sentences in the generated summaries to match the reference summaries might improve the performance significantly.

References

[1] Devlin, J. & Chang, M. W. & Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[2] Radford, A. & Wu, J. & Child, R. & Luan, D. & Amodei, D. & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.

[3] Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

[4] Nallapati, R. & Zhou, B. & Gulcehre, C. & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

[5] Kim, B. & Kim, H. & Kim, G. & (2018). Abstractive summarization of reddit posts with multi-level memory networks. *arXiv preprint arXiv:1811.00783*.

[6] Radford, A. & Narasimhan, K. & Salimans, T. & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[7] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74-81).

[8] dmmiller612. (n.d.). *DMMILLER612/bert-extractive-summarizer: Easy to use extractive text summarization with bert*. GitHub. Retrieved April 16, 2023, from https://github.com/dmmiller612/bert-extractive-summarizer

[9] Miller, D. (n.d.). *Bert-extractive-summarizer*. PyPI. Retrieved April 16, 2023, from https://pypi.org/project/bert-extractive-summarizer/

[10] *CNN-dailymail news text summarization*. Kaggle. (n.d.). Retrieved April 16, 2023, from https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail

[11] *Reddit_tifu: tensorflow datasets*. TensorFlow. (n.d.). Retrieved April 16, 2023, from https://www.tensorflow.org/datasets/catalog/reddit_tifu#reddit_tifulong

Contributions

Xi Zhang: Found related work and resources, wrote all the code, ran models on the Reddit TIFU dataset, created model architecture graph, wrote the entirety of this paper
Emre Cagin: Proposed project topic, ran models on the CNN-Daily Mail dataset, gave feedback on this paper

6 Optional Appendix

6.1 Embedding Mechanisms

6.1.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language model that is commonly used for generating word embeddings. As the name suggests, BERT uses bidirectional attention, which is able to better capture the full context of a sentence than unidirectional attention which is only capable of attending to the preceding context. BERT is pre-trained using two unsupervised tasks, namely masked language modelling (MLM) and next sentence prediction (NSP) [1]. In MLM, BERT takes a sentence and randomly masks some of its tokens, which the model attempts to predict based on the remaining unmasked tokens in the sentence. This technique forces the model to learn how to generate contextually relevant word embeddings. In NSP, BERT takes two sentences, sentence A and sentence B, and tries to predict whether sentence A directly precedes sentence B [1]. This allows the model to learn inter-sentence relationships.

6.1.2 GPT-2

GPT-2 (Generative Pre-trained Transformer 2) is a multi-layer decoder-only transformer which uses unidirectional attention. During pre-training, the model is fed a sequence of text tokens in which the model tries to predict the next tokens based on the preceding tokens. As a result, GPT-2 is better suited for tasks that involve generating coherent text. Just like BERT, GPT-2 can also be used to learn contextual embeddings and be fine-tuning to perform various downstream NLP tasks [2, 6].

6.2 K-Means Clustering

K-Means Clustering is an unsupervised algorithm aimed at partitioning N number of obserations into K clusters based on feature similarity. In the context of text summarization, these K clusters are analogous to the K groups of sentences grouped based on their syntactic and semantic properties.