Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220520

Abstractive summarization of COVID-19 with transfer text-to-text transformer

Zhaopu Teng

Boston University, Boston MA 02215, USA

zhaoput1@bu.edu

Abstract. As a classic problem of Natural Language Processing, summarization provides convenience for studies, research, and daily life. The performance of generation summarization by Natural Language Processing techniques has attracted considerable attention. Meanwhile, COVID-19, a global explosion event, has led to the emergence of a large number of articles and research. The wide variety of articles makes it a perfect realization object for summarization generation tasks. This paper designed and implemented experiments by fine tuning T5 model to get an abstract summarization of COVID-19 literatures. A comparison of performance was shown to prove the reliability of the model.

Keywords: transfer learning, natural language processing, COVID-19, attention-based model, summarization, text generation.

1. Introduction

When an international explosion event happens, the relevant articles or research will grow exponentially. People need to spend time on reading them and finding out the key points that they are interested in. It could be very time-consuming tasks especially for those without abstracts or highlights.

We noticed that the articles related to COVID-19 became massive suddenly from 2020 after the pandemic outbreak. The articles include academic papers, news reports, posts on twitter or other social media. Academic papers usually have an abstract to help readers understand the whole passage before they jump into it. However, in news reports or long posts on social media, the abstract parts are usually missing. Hence, to extract the important messages, spending much time on them is needed. But emergencies can change and develop rapidly, and the blow-out of corresponding articles will be hard to follow. Under these circumstances, the summarization becomes crucial.

Natural language process (NLP) has progressed greatly in recent years. At the beginning, word2vec can describe the whole passage, and then Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) come out with incredible experimental results. In the last three years, with the popularity of Attention and Transformer [1], BERT [2] came out in 2018, providing us with a superb solution to a variety of NLP problems. In addition, those technologies have been used in various areas, such as automation translation, sentiment analysis for reviews or providing sales suggestions, articles classification, and etc. I believe that it could also help us on the task of summarizing articles.

There are two types of summarization in NLP: extractive summarization and abstractive summarization. The extractive summarization is to identify important information in the passage and

DOI: 10.54254/2755-2721/2/20220520

regroup them together as an output. On the other hand, abstractive summarization summarizes the content with created phrases which can be seen as paraphrasing.

New findings have extraordinary performances in many NLP areas which rapidly grow in recent years and Sentence BERT [3] was built on those and provided a good example to process long sequence NLP model. Further, there are a number of extractive summarization papers that give out good results by utilizing BERT and fine-tuning it. Also, continual BERT [4] was utilized as well on the COVID-19 papers form PubMed and ScisummNet dataset. In July 2020, Google launched the T5 model [5] which provides another model for text-to-text tasks.

In this paper, I am going to talk about related work in transformer and both types of summarization in the second section. In the third section, I will build up a T5 model, apply on COVID-19 literature and fine-tune to achieve a better performance in summarization tasks. ROGUE-1 F, ROGUE-2 F, ROGUE-L will be used to evaluate the result. This paper will then discuss about hyperparameters and beam search with different beam width and length penalty. A comparison will be provided between T5 and other models at the end of this paper.

2. Related work

2.1. Transformer

As a milestone of NLP, Attention was firstly proposed in 2015 [6] which tried to improve the performance of the Seq2Seq model, since the final state of Seq2Seq model built by RNN could lose information due to a long sequence. However, Attention could help the decoder review all the states of the encoder to avoid forgetness and emphasize the certain encoder state to let decoder focus on. Later, Transformer was proposed in 2017, trying to only utilize multi-head attention and dense layers, which had an incredible performance and became one of the start-of-art models in NLP. BERT is a pre-trained model built on top of Transformer, proposed by a google team in 2019, which has an astonishing result. There are two pre-train processes for the encoder which are Masked Language Model and Next Sentence Prediction to make every page possible to be a training dataset. T5 is another milestone in solving text-to-text problems such as translation, text regression, summarization, which is a standard encoder-decoder transformer.

2.2. Summarization

2.2.1. *Extractive summarization.* Extractive summarization is identifying the most important sentence in a paragraph, and rearranging them together to produce the summarization, which can be regarded as a subset of the passage with highest weight. SummaRuNNer [7] proposed an interpretable sequence model based on RNN and made the result visualization. Shashi Narayan, etc. applied Reinforcement Learning (RL) on the extractive summarization task [8] by ranking the sentences. BertSum [9] applied Transformer on to the summarization with multiple layers. Neural models handle extractive summarization more like a sentence classification problem that use a classifier to predicts which sentence should be the summary which one is not. These methods can provide good test scores and high-quality results for extractive summarization, proved that it is a good approach for solving summarization tasks.

2.2.2. *Abstractive summarization.* Instead of selecting the most important sentences, abstractive summarization aims to understand the paragraphs and generate a short abstraction itself. Romain Paulus, etc. [10] used a new training method by combining word prediction and RL to solve longer document abstractive summarization. Discourse-Aware [11] proposed the abstractive summarization model with an attentive discourse-aware decoder to generate the abstraction. A sequence-to-sequence model is usually used to solve the abstractive summarization, which applies an encoder to map source text to vectors, and the model will provide a calculated vector as a result followed by a decoder restoring the vector to the summarization.

DOI: 10.54254/2755-2721/2/20220520

Comparing with extractive summarization, although both of them can provide a similar result evaluation and could both be a good solution to summarization problem, the internal processes of the two methods are different. Abstractive summarization is more like machine paraphrasing after getting whole idea of paragraph rather than combining important sentences together, in a way similar to what human do when we come up with a summary. Hence, this paper will focus on abstractive summarization, and transformer based T5 has an overwhelming advantage on long-sequence text generation problems which can be applied on abstractive summarization tasks.

3. Methodology

I tried to use the supervised learning techniques on summarization of COVID-19 literature. COVID-19 related papers can be found in PubMed and Kaggle, and the abstracts can be used as the golden standard of the summarization. This paper was aimed to build a sequences-to-sequences model from scratch by preprocessing raw data, setting up data module, utilizing pretrained T5, fine tuning with all the parameters to generate a model for COVID-19 literature abstractive summarization tasks.

3.1. Model architecture

To solve text to text questions, Google pretrained the T5 model. There are three structures that are frequently used, Encoder-Decoder, Language model, and Prefix LM. T5 chose the Encoder-Decoder transformer structure (see figure 1) that got the best performance from their experiment with natural language generation tasks. And the attention mask pattern they chose was a fully visible mask where all the output entries were able to check the status of all the input entries. To pretrain the model, a BERT style mask was utilized with 15% and along with a Replace span mask strategy with length 3.

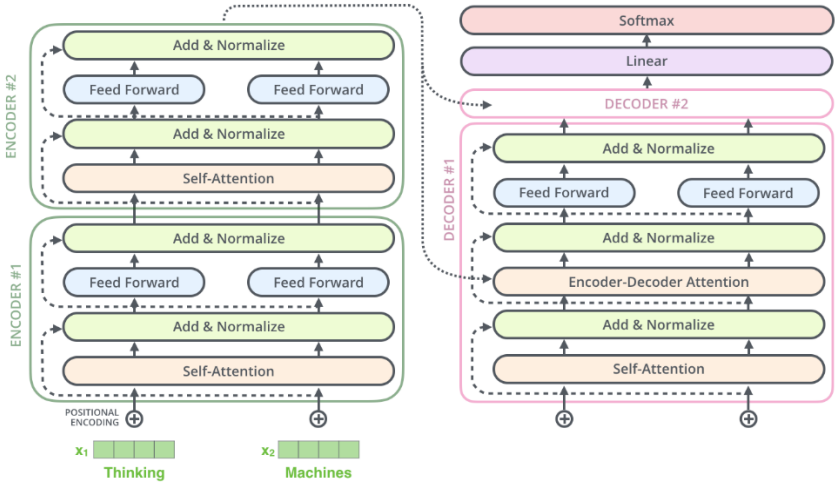

Figure 1. Architecture of T5 model [12].

In this paper, I am going to use t5-base and fine tune it to get a good performance on COVID-19 dataset. The encoder and decoder of T5-base consist of 12 blocks, which has approximate 220M parameters 3072 feed forward hidden state, 768 hidden state, 3072 feed forward hidden state, 12 heads. Additionally, T5 tokenizer is used to do the word embedding which is based on SentencePiece. SentencePiece is an unsupervised text tokenizer used for text generation tasks and the vocabulary size is predetermined before the model training.

3.2. Data preparation

The dataset used is COVID-19 Open Research Dataset from Kaggle which has 801,822 literatures prepared by White House and a coalition of leading research groups, which recruits COVID-19, SARS-CoV-2, and other literatures about corona viruses. Since the abstract is needed as the golden standard for the prediction, the literatures with no abstracts and corresponding files are removed. 183658 literatures remained in the dataset, and the token count for the dataset is shown in figure 2. Some of the

DOI: 10.54254/2755-2721/2/20220520

literatures have extremely long tokens which text tokens could be larger than 50k, and for abstract token size larger than 1k. A data cleaning was done to filter them out, since this paper only focused on the text token size less than 5k and summary token size less than 500, and finally got 69598 literatures (see figure 3). A train-test-validation split was set up as 9:1:1 for this experiment.

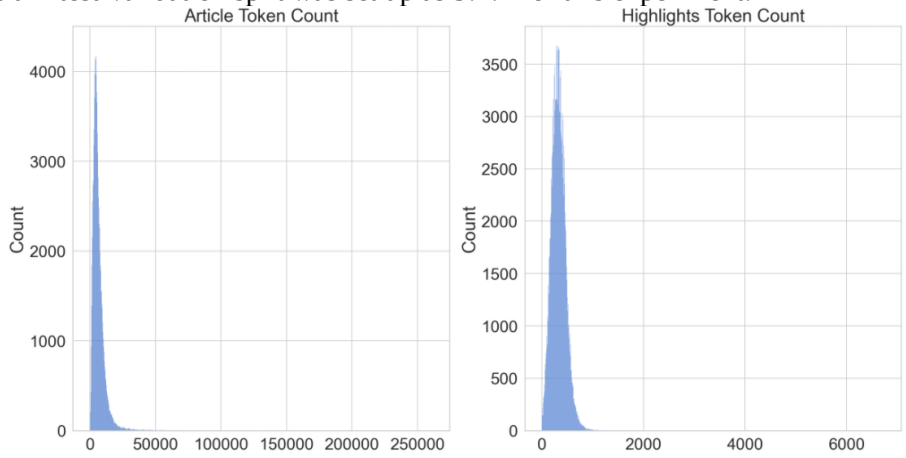

Figure 2. Distribution of cov-19 dataset token.

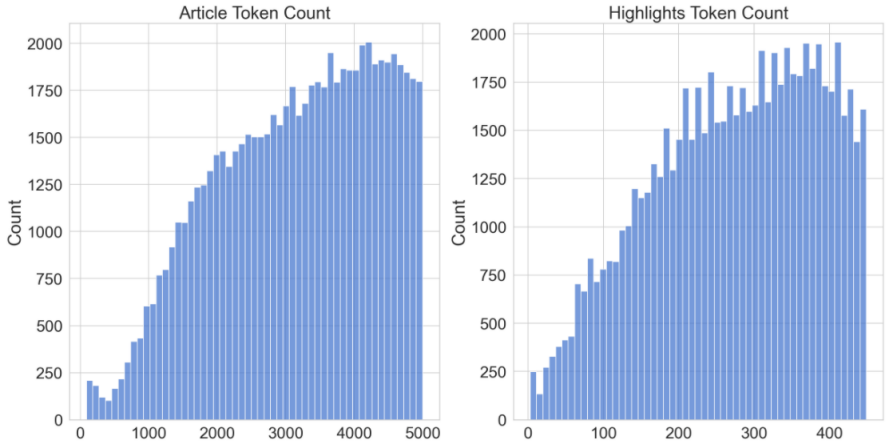

Figure 3. Distribution of processed cov-19 dataset token.

3.3. Experiment setup

Pytorch lightning and Pytorch are utilized in the model building, training, and testing. The dataset is defined with 512 or 1024 text max token length, 200, 250 or 300 summary max token length, and will truncate the data if the token size is larger than that. The training has 3 to 5 epochs in total and the batch size is set to be 4 for text token size is 1024, and 14 for text token size is 512. AdamW optimizer is utilized and both fixed learning rate and a dynamic learning rate are used to train the model.

4. Result

4.1. Training steps

During the training, the train loss is recorded every 250 steps and do the validation check every 10% epoch and get the graph below (see Fig. 4). Checkpoints are used to record the best model and get the best model by loading from the checkpoint with lowest validation loss.

DOI: 10.54254/2755-2721/2/20220520

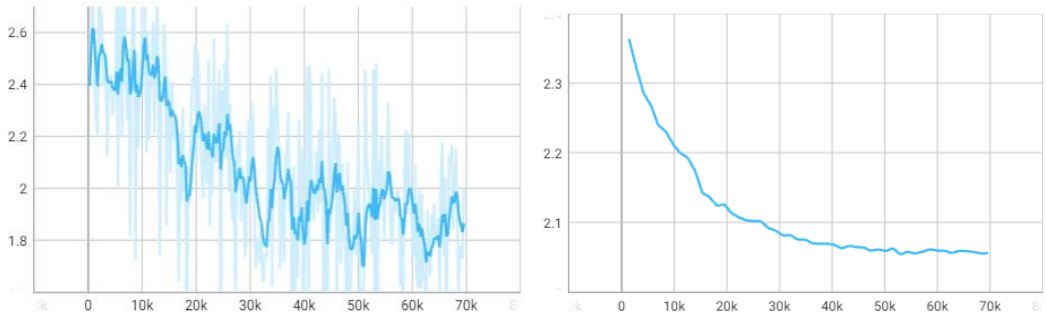

Figure 4. Train/Validation loss.

4.2. Beam search

Beam Search is a Heuristic search algorithm that explores a graph by expanding the most promising node in a limited set. It is frequently used in text generation tasks. Although the parameter for the T5 model is beam width of 4 and a length penalty of 0.6 (Raffel, et al., 2019), the COVID-19 dataset has a different token size in both text and summary part which could affect the beam width and length penalty chosen. A series of experiments were designed to explore if another combination of beam width and length penalty was needed.

Since the token size in COVID-19 dataset is significantly larger than CNN/DailyMail dataset (see figure 5), and the beam search experiment is a very time-consuming task, 25% of the test datasets are used to excise the experiment and get the Table 1 as following. Lower length penalty and large Beam width will decrease the scores. And the result shows that a beam width of 4 and a length penalty of 0.6 will provide the best performance in COVID-19 dataset as well.

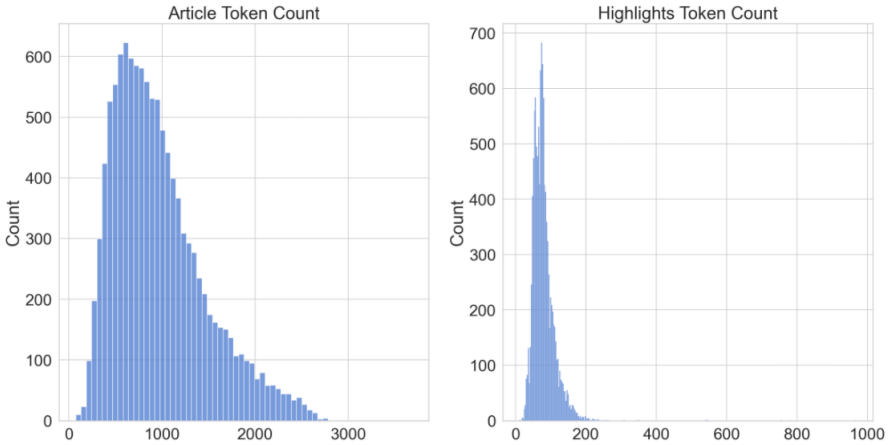

Figure 5. distribution of CNN/DailyMail dataset token.

Table 1. Beam width vs length penalty.

Length Penalty		Beam=3	Beam = 4	Beam = 6	Beam = 8
0.2	Rouge-1 F	0.3201	0.3160	0.3083	0.3020
	Rouge-2 F	0.1283	0.1252	0.1218	0.1188
	Rouge-L F	0.2929	0.2884	0.2811	0.2755
0.4	Rouge-1 F	0.3238	0.3139	0.3085	0.3206
	Rouge-2 F	0.1311	0.1236	0.1225	0.1237
	Rouge-L F	0.2960	0.2857	0.2812	0.2791

DOI: 10.54254/2755-2721/2/20220520

Table 1. (continued).

0.5	Rouge-1 F	0.3147	0.3106	0.3122	0.2898
	Rouge-2 F	0.1248	0.1207	0.1281	0.1259
	Rouge-L F	0.2869	0.2833	0.2856	0.2823
0.6	Rouge-1 F	0.3198	**0.3246**	0.3155	0.3111
	Rouge-2 F	0.1296	**0.1272**	0.1277	0.1251
	Rouge-L F	0.2936	**0.2970**	0.2884	0.2841
0.7	Rouge-1 F	0.3181	0.3240	0.3193	0.3144
	Rouge-2 F	0.1266	0.1290	0.1275	0.1269
	Rouge-L F	0.2898	0.2957	0.2909	0.2877
1	Rouge-1 F	0.3226	0.3228	0.3208	0.3212
	Rouge-2 F	0.1282	0.1303	0.1304	0.1308
	Rouge-L F	0.2951	0.2955	0.2934	0.2927

4.3. Model evaluation

The best performance model got the smallest validation loss at 2.054 with text token size equals to 1024, summary token size equals to 300. Dynamic StepLR is used and started at lr = 5e-4, and gamma=0.5 (see figure 6). Beam width is 4 and length penalty $\alpha = 0.6$, repetition penalty is 2.5.

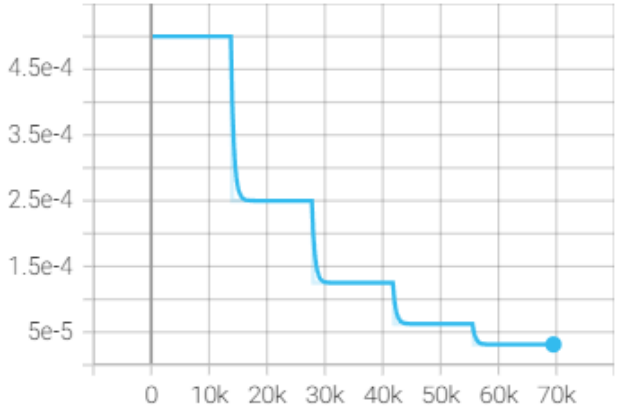

Figure 6. Learning rate vs steps.

And the evaluation results are shown in Table 2. T5-Base has 222M parameters less than BART-Large(406M) and PEGASUS-Large (568M) but provide a significantly higher Rouge F score than others. Although BART-Large provides higher Recall on Rouge-1 and Rouge-L, the summary machine generated should not only need to ensure we get all the important information but also have the possibility to filter out alternative messages.

Table 2. Model evaluations.

	Rouge-1			Rouge-2			Rouge-L		
Model	F	P	R	F	P	R	F	P	R
T5-Base	**0.3246**	**0.3178**	0.3638	**0.1272**	**0.1233**	**0.1479**	**0.2970**	**0.2906**	0.3330
BART-Large	0.2051	0.1526	**0.3664**	0.0597	0.0431	0.1242	0.1873	0.1390	**0.3357**
PEGASUS-Large	0.2712	0.3006	0.2847	0.0965	0.1095	0.1035	0.2465	0.2733	0.2591

DOI: 10.54254/2755-2721/2/20220520

5. Conclusion and discussion

In this paper, a T5 model was implemented to fulfill the COVID-19 literature summarization task. It is found that a larger max token size will not only improve the performance of the model but also increase the cost of the calculation. A balance, whether we need a higher accuracy or less time cost, should be made during different tasks. But for the combination of beam width and length penalty for the best model stayed consistent. T5-base already has a good performance in the COVID-19 literature summarization task. In the future, I can further investigate a large size model, e.g., T5-large, and its performance on longer passages.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[2] 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.

[3] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[4] Park, J. W. (2020). Continual bert: Continual learning for adaptive extractive summarization of covid-19 literature. arXiv preprint arXiv:2007.03405.

[5] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

[6] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[7] Nallapati, R., Zhai, F., & Zhou, B. (2017, February). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence.

[8] Narayan, S., Cohen, S. B., & Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636.

[9] Liu, Y. (2019). Fine-tune BERT for extractive summarization. arXiv preprint arXiv:1903.10318.

[10] Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304.

[11] Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. arXiv preprint arXiv:1804.05685.

[12] Alammar, J. (2018, June 27). The Illustrated Transformer. From http://jalammar.github.io/illustrated-transformer/