Project Report : CS 7643 O1 - Fall 2020
Parameter-Efficient Adapter-Based BERT for Extractive Summarization

Soroush Bassam
Georgia Institute of Technology
sbassam3@gatech.org

Abstract

Fine-tuning, the standard process in which a pre-trained language model is re-trained on a supervised task such as question answering, can often be costly. Adapter-training offers a lightweight alternative to fine-tuning while providing extensibility. Previously, the method showed promising results on the GLUE benchmark. This project utilizes adapter modules for the canonical summarization task, namely the CNN/Daily Mail dataset. The effectiveness of the adapter modules are then compared to the standard fine-tuning practice. The experiments reveal some issues related to the model generalizability as well as providing some insight into the effect of summary length on the metric values. Project repository can be found on github. [1]

1. Introduction

Summarization is an important and challenging task within Natural Language Processing (NLP). Recently, with the advent of Transformers, several models have achieved State-of-the-art results[2]. The current practice in tackling these tasks involves fine-tuning a pretrained language model on a downstream supervised summarization dataset. This is often costly as all the parameters of the network are impacted. Adapter-based models, however, perform reasonably well with significantly fewer parameters compared to regular transformers [2]. Houlsby et al [2] proposed an adapter-based model that used 3.6% of the parameters used by a conventional fine-tuning model while attaining within 0.4% of its performance. This project proposes a similar adapter-based approach for summarization. The goal is to implement adapters for the canonical CNN/Daily Mail dataset and study their effectiveness as compared to the standard fine-tuning approach. The rest of this section will provide more context around the research objective.

There are two approaches to summarization. The Extractive approach directly copies the essential sentences from the original document. The Abstractive approach, on the other hand, may use new words to paraphrase the source document [1]. The goal in either case is to condense a document and collect only the essential information.

CNN/Daily Mail dataset that is quintessential in summarization research. Originally collected by Hermann et al., the dataset is a set of over 300,000 news stories written by CNN/Daily Mail journalists [3]. Each story is paired with a reference summary that will be the basis for the model's performance evaluation.

As described in section 3, the adapter training method shows promising results in the context of summarization. When evaluated on a subset of the dataset, despite potential overfitting issues, the adapter-based model still outperforms the fine-tuned model on a subset of the data. If we can generalize this result to the entire dataset, there will be a significant reduction in both training time as well as the number of parameters used. From a practical point of view, fewer parameters translates into faster and cheaper training. Also, since adapters provide a high degree of parameter sharing [2], the parameters from the model trained on a summarization task can then be used on a different task, such as question answering.

Adapterhub [4], the library that was built on top of the transformers library to support adapter modules, has not yet added a summarization task. This project aims to contribute to the library by defining the CNN/Daily Mail summarization task as well as uploading the trained adapters for public access.

2. Approach

This section describes the model architectures, the training, validation, and testing processes as well important implementation details.

There are several considerations for choosing a summarization method. For instance abstractive methods require

[1] Project repo: https://github.com/sbassam/cs-7643-project.git

[2] Top models http://nlpprogress.com/english/summarization.html

[3] https://github.com/huggingface/datasets/tree/master/datasets/cnn$_{dailymail}$

[4] https://adapterhub.ml/

a decoder and a more complex architecture. On the other hand, the extractive summarization can be thought of as a binary text classification problem where each sentence is labeled as important or disposable. Therefore, it only requires an encoder and a classifier at the minimum. Since seq2seq tasks requiring a decoder are not yet supported by the adapter-transformers library, and also due to hardware limitations, the extractive method was chosen.

The project setup allows for comparison between a fine-tuned model and an adapter-based model. In order to make a fair comparison, they are both based on the same language model, namely BERT-base. Furthermore, both models use a similar linear classification head. Due to the large size of CNN/Daily Mail dataset, a sample of about 5-10% of the entire dataset was used. As discussed in the next section, this turns out to affect both models' performances so in order to control for this factor a baseline model performance is listed for comparison. Developed by Liu et al., BERTSUM is similar to the fine-tuned model except it learned from the entire CNN/Daily Mail training split as opposed to the fine-tuned model developed for this project (which was trained on only 10% of the data).

The fine-tuned model uses BERT-base from the huggingface library[5]. A simple linear classifier was used as a head for the downstream sentence classification task. The adapter-based model utilizes the BERT model from the same library but with a sequence classifier already implemented on top. Then using the adapter-transformers library, a single adapter module is added for and all the original BERT weights are frozen during training.

While these models use the same data for training, validation, and testing, the adapter model required the data to be broken down into sentences since the adapter modules showed limited flexibility if ingesting multiple [CLS] tokens. As discussed in the next section, one anticipated problem was a loss of contextual information as sentences are being learned individually. But it turns out despite this shortcoming, the adapter-based model showed superior results.

The new approach introduced in this project combines two ideas that were recently deemed successful. Liu et al. [4] first showed that inserting [CLS] tokens for each sentence in a document in addition to segment embeddings achieves successful extractive performance. But this idea wasn't implemented using adapter modules. Also as previously mentioned, adapter modules have not been evaluated on summarization tasks, therefore this project will examine potential issues that may arise when these two ideas are combined.

This approach resulted in promising performance for the adapter based model. While there is a gap between the adapter-based model's current performance and the State-of-the-art, the yet larger gap between the fine-tuned model and BERTSUM [6] indicates that when trained on the entire set, the adapter-based model could achieve superior performance while using only a fraction of parameters.

3. Experiments and Results

The metrics were chosen with the goal of enabling comparison to existing models. Recall-Oriented Understudy for Gisting Evaluation, or ROUGE, are a family of metrics that are commonly used to measure the performance of models on summarization tasks. These metrics compare the summaries generated by to the model to a set of reference summaries. The four metrics from this family that are usually reported are: ROUGE-1, ROUGE-2, and ROUGE-L which correspond to the overlap of unigrams, bigrams, and longest sequences of words, respectively. For the purposes of this project, the average F-measure of each metric was reported.

As described in the previous section, both the fine-tuned and the adapter-based models were trained on a subset of about 10% of the CNN/Daily Mail dataset. Then each model was validated against the same proportion of the validation dataset. Finally, after tuning some of the hyperparameters such as the reduction factor, the models were tested against 5% of the testing dataset. The results are listed in Table 1. The adapter-based model achieved slightly better result with using only about 0.8% of the parameters while the fine-tuned approach engaged all parameters. The results are an indicator of the approach's effectiveness as well as efficiency. This aligns with the findings by Houlsby et al. [2] and Gururangan et al. [3] on other tasks such as Question Answering.

As indicated in the adapter-based model learning curve plot, the validation set is showing an increase in loss while the training performance is improving with each epoch. That's a clear sign that the adapter-based model is suffering from overfitting. The fine-tuned model on the other hand, is showing that the validation curve is slowly improving which could be a sign that the model has more learning capacity but limited available data. For the purposes of this project, the focus is on studying the adapter-based model. So the following experiments are set to accomplish this goal.

In the first experiment, the dropout rate was chosen as the parameter of interest to study its effect on overfitting. The adapter model ran against 10% of the validation dataset with the dropout ranging from 0.01 to 0.85. First the dropout probability for all fully connected layers in the embeddings, encoder, and pooler was used. The same experimental setup with dropout ratio for the attention probabilities showed similar results. As depicted in Figure 2, changing the either dropout ratios had an impact on the training performance (e.g. dropout ratio of 0.1 attains the best performance) but

[5] https://huggingface.co/transformers/index.html

[6] https://github.com/nlpyang/PreSumm/issues

Model	ROUGE-1	ROUGE-2	ROUGE-L	No. of Parameters
Adapter-based BERT	0.37	0.15	0.23	896,066 (0.8%)
Fine-tuned BERT	0.34	0.14	0.21	109,483,009 (100%)
BERTSUM+Classifier	43.23	20.22	39.60	109,483,009 (100%)

Table 1. Performance comparison between the dataper-based model, fine-tuned model, and BERTSUM

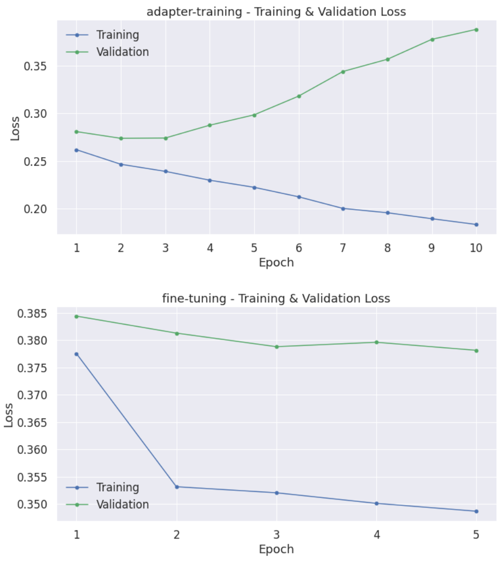

Figure 1. Learning curve for adapter-based as compared to the fine-tuning method prior to experiments

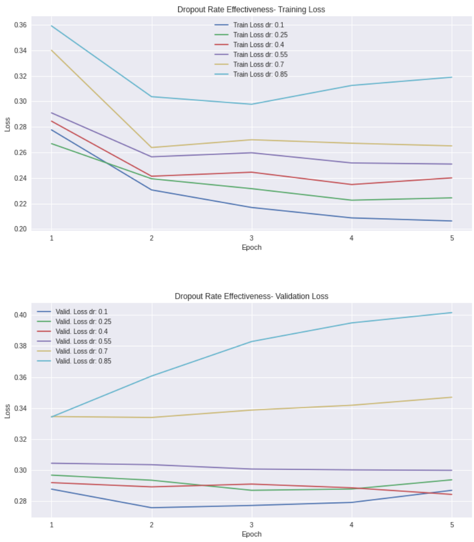

Figure 2. The effect of dropout on adapter-based model's learning performance

limited effect in avoiding overfitting. Conducting other experiments to avoid overfitting showed similar results. Some of the parameters that were used includes the reduction factor. A more thorough list of experiments can be found in the repository.

The conclusion from this experiment was that even though the adapter-based model turns out to outperform the fine-tuned model, tuning individual hyper-paramaters is not a remedy for the lack of proper generalizability. This could indicate that a more comprehensive change (such as an architectural modification) is needed to add more regularization.

The second experiment concerns the length of the generated summary. The goal here is to examine the effect of summary length on the performance metrics. The hypothesis is that summaries that are too short cannot convey the essential information. Moreover, summaries that are too long may convey the essential information, but they also contain less important sentences. the expectation is that in both of these cases, all the metrics be lower and there would be an optimum in the middle. This experiment is set to find that point.

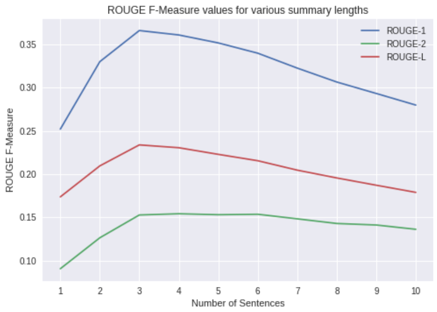

Figure 3. The role of summary length- Summaries containing three sentences yielded the highest F-Measure in all three metrics

The adapter-based model that was previously saved after training was used for this experiment. The length of the summaries varied from a single sentence up to 10. Figure 3 shows the result of running the model on approximately 5% of the validation set. Interestingly, all three metrics reach an optimum at the same length. This was unexpected as different ROUGE metrics measure different qualities of a summary. ROUGE-1 tends to prefer summaries with more overlapping unigrams which could mean that shorter summaries with exact words get higher values. By contrast, ROUGE-L tends to bias toward summaries with longer sequences. But in the context of extractive summarization this makes more sense since in this approach the exact sentences are retrieved so it's likely that a sentence that gets selected has too much of a weight in the overall performance, hence fewer sentences could mean better results.

Both models were trained on about 10 percent of the CNN/Daily Mail dataset. While the preliminary results expectedly under-perform state-of-the-art models such as Presumm. From a qualitative viewpoint, using only a fraction of the data potentially contributes to this discrepancy. More importantly, the two models (adapter-based and fine-tuned) show similar performance. In fact, the adapter-based model outperforms the fine-tuned model in all three metrics. This comes as a surprise since the former model only engages about 0.8% percent of the latter model during training.

4. Limitations & Future Work

Several limitations have already been mentioned, but this section maps some of the more important issues to potential future work venues.

The most obvious flaw in the adapter-based model was that even though it shows better results than the fine-tuned model, it suffers from overfitting. The three experiments with the dropout ratio and adding normalization layers to the adapters did not improve the model's ability to generalize. One potential venue for the future work is to train these models on a larger portion (ideally all) of the data and examine how the model performance changes.

Another aspect of this research that needs further examination is the architecture of the adapter modules. As mentioned earlier, one benefit of the adapter modules is their extensibility and high degree of parameter sharing. It is interesting to examine whether the model will be able to achieve a better generalizability by changing the architecture or fusing several existing architectures.

References

[1] P. J. Liu A. See and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017. 1

[2] N. Houlsby et al. Parameter-efficient transfer learning for nlp. *arXiv:1902.00751*, 2019. 1, 2

[3] S. Gururangan et al. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv:2004.10964*, 2020. 2

[4] Y. Liu and M. Lapata. Text summarization with pretrained encoders. *arXiv:1908.08345*, 2019. 2