SQuAD 2.0 Question Answering with Reformer

Stanford CS224N Default Project

Haiyuan Mei
Department of Computer Science
Stanford University
hmei0411@stanford.edu

1 Key Information to include

- External collaborators (if you have any): NA
- Mentor (custom project only): NA
- Sharing project: NA

2 Research paper summary (max 2 pages)

Title	Attention Is All You Need
Venue	Neural Information Processing Systems Foundation
Year	2017
URL	https://arxiv.org/abs/1706.03762

Table 1: Attention Is All You Need [1].

Title	Reformer: The efficient transformer
Venue	International Conference on Learning Representations (ICLR)
Year	2020
URL	https://arxiv.org/abs/2001.04451

Table 2: Reformer: The efficient transformer [2].

Background. Attention based Transformers based models now achieve state of the arts results in many NLP tasks. However one big problem of about these models is it's size and requirements for computation. One example is that Google made use of 16 TPUs in order to train the pretrained BERT model; which made it very difficult for individual researchers to work in this area.

There are various models recently introduced to mitigate the complexity problem, and Reformer is one of the models that can be used to reduce the size of the model and speed up the training of attention based transformers.

Summary of contributions. Attention Is All You Need [1] is the original paper which introduced the transformer model. It is the foundation of almost all of today's state of the art language models, including but not limited to BERT, XLNet, ALBERT, etc. It introduced to the NLP world a brand new deep network, outperforms the previous generation RNN based models, and will be likely the most popular research topic in the near future.

Reformer: The efficient transformer [2] is a very recent work aimed at increasing the speed of transformer model. It is designed to handle 1 million words in context windows, while using only

Title	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Venue	
Year	2018
URL	`https://arxiv.org/abs/1810.04805`

Table 3: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [3].

16GB of memory. It combines two techniques: locality-sensitive hashing to reduce the sequence-length complexity as well as reversible residual layers to reduce storage requirements.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [3] - by training deep Transformers on a carefully designed bidirectional language modeling task, BERT achieves state-of-the-art results on a wide variety of NLP benchmarks including SQuAD 2.0. However in order achieve state of the art result using BERT, Pre-training is fairly expensive (four days on 4 to 16 Cloud TPUs).

Limitations and discussion. One major issue with transformer based models is the efficiency on long sequences with limited memory capacity. Various methods are introduced to mitigate this issue: TransformerXL introduces recurrence on vanilla Transformers; ALBERT achieves this by allocating the model's capacity more efficiently and sharing parameters across the layers; Reformer is one such model that could be used to solve the problem of long sequence with limited memory by using locality-sensitive hashing and reversible residual layers. Another problem with Transformers/BERT models is that it only can use forward context or backward context, which means it can't use forward and backward context at the same time. XLNet is yet another transformer based model, by applying Permutation Language Modeling, which helps to outperform BERT model.

Why this paper? In this course project I will mainly focus on two papers: Attention Is All You Need [1] and Reformer: The efficient transformer [2]. As the main part of this project, a replication of Reformer model will be implemented, and will be compared to the existing Transformer model to study their differences and performance (The transformer model is now available in pytorch: `https://pytorch.org/tutorials/beginner/transformer_tutorial.html`) Further steps will be to explore Transformer based pre-trained models such as BERT, ALBERT, XLNet or TransformerXL, which have gained immense success in NLP related tasks. It would be very interesting to see if a different attention mechanism as introduced by Reformer could be used to increase the performance of BERT and achieve single GPU pre-training.

Wider research context. Transformer models are a new family of NLP models, which outperforms RNN based models in a lot of NLP tasks; especially in language representation, as has been proved that the BERT/ALBERT pre-trained model helps to outperform human performance. In the foreseeable future transformer models will be a very popular topic in NLP study.

3 Project description (1-2 pages)

Goal. In this course project, firstly I will be implementing and evaluating a neural model-based system, the Reformer model [2], using pytorch for the question-answering tasks defined in SQuAD2.0 [4]; the project will be non-PCE based, confining the study on the Reformer model and the original Transformer [1] only. Additionally, after that the model reaches the goal of outperforming the baseline BiDAF model, I will continue exploring whether the Reformer model will achieve similar result as the original Transformer in SQuAD2.0 task while requires much less computation.

Furthermore, the two techniques: locality-sensitive hashing and reversible residual layers in Reformer looks to be a very promising method in improving BERT model, this will be my stretch goal.

Task. Reformer [2] based question-answering tasks defined in SQuAD2.0 using pytorch.

Data. As the main part of the project is non-PCE based, SQuAD 2.0 training and dev set will be used to train the model. To be more specific, the training set contains 129,941 examples all taken from the official SQuAD 2.0 training set, and the dev set contains 6078 example, roughly half of the official dev set, randomly selected.

Methods. Describe the models and/or techniques you plan to use. If it's already described in the paper summary, no need to repeat. If you plan to explore a variant to a published method, focus on describing how your method will be different. Make it clear which parts you plan to implement yourself, and which parts you will download from elsewhere. If there is any part of your planned method that is original, make it clear.

Baselines. Two baselines will be used in this project. The default course baseline model is BiDAF model without character-level embedding layer; The baseline is already implemented and provided by CS224n teaching group, and previous published F1 and EM scored will be used to be compared with the Reformer model results. A more interesting baseline will be the pytorch transformer as provided in the library; a transformer model will be trained and F1/EM scores generated to be compared with the Reformer model; furthermore, I will also compare the model size and the training speed of transformer and reformer models.

Evaluation. After the experiment, I will be using F1 and EM scores to evaluate the quality of the new Reformer model (compared to the two baselines). I will also be comparing the size of Transformer and Reformer, the training speed and the computation resource required by each of the models.

References

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

[2] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[4] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018.