

Multi-task Learning with Multi-head Attention for Multi-choice Reading Comprehension

Hui Wan

IBM Research AI
hwan@us.ibm.com

Abstract

Multiple-choice Machine Reading Comprehension (MRC) is an important and challenging Natural Language Understanding (NLU) task, in which a machine must choose the answer to a question from a set of choices, with the question placed in context of text passages or dialog. In the last a couple of years the NLU field has been revolutionized with the advent of models based on the Transformer architecture, which are pretrained on massive amounts of unsupervised data and then fine-tuned for various supervised NLU tasks. Transformer models have come to dominate a wide variety of leader-boards in the NLU field; in the area of MRC, the current state-of-the-art model on the DREAM dataset (see [Sun *et al.*, 2019]) fine tunes Albert, a large pretrained Transformer-based model, and additionally combines it with an extra layer of multi-head attention between context and question-answer [Zhu *et al.*, 2020]. The purpose of this note is to document a new state-of-the-art result in the DREAM task, which is accomplished by, additionally, performing multi-task learning on two MRC multi-choice reading comprehension tasks (RACE and DREAM).

1 Introduction

Making a computer system understand text within context and answer questions is challenging but has attracted a lot of interest of the Artificial Intelligence community and general audience for a long time. In the recent years, many Machine Reading Comprehension (MRC) datasets have been published, with different genres and formats of the context and questions. The context could be in the form of text passages, or in the form of dialogues. The questions could be open-formed (e.g. HotPotQA [Yang *et al.*, 2018]), asking the system to either extract the answers as spans from the context or external knowledge, or abstract and summarize the answers; the questions could also be in the form of asking the system to choose the best answer from multiple choices. In this note we will focus on the multi-choice MRC tasks, more specifically, the DREAM task [Sun *et al.*, 2019].

1.1 Datasets

RACE [Lai *et al.*, 2017] is a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions. The average passage length is 322 words. Each question provides 4 answer options to choose from. The human ceiling performance is 94.5.

DREAM [Sun *et al.*, 2019] is a much smaller reading comprehension dataset with more than 6,000 dialogues and over 10,000 questions. The average dialogue length is 86 words. Each question provides 3 answer options to choose from. The human ceiling performance is 98.6.

1.2 Related Work

Early works on the DREAM task include feature-based GBDT [Sun *et al.*, 2019], and FTLM [Radford *et al.*, 2018] which is based on the Transformer [Vaswani *et al.*, 2017] architecture. The top system accuracy on the DREAM leaderboard has been advanced gradually to above 90 percent, since the break-through of the text encoder in the form of large pretrained Transformer-based models (BERT [Devlin *et al.*, 2019], XLNet [Yang *et al.*, 2019a], RoBERTa [Liu *et al.*, 2019], Albert [Lan *et al.*, 2020]).

Transfer learning is a widely used practice in machine learning (ML) that focuses on utilizing knowledge gained while solving one problem and applying it to a different but related problem. Using pretrained language models (LMs) such as ELMO [Peters *et al.*, 2018] and BERT [Devlin *et al.*, 2019] in down-stream tasks is an example of sequential transfer learning. On the other hand, multi-task learning, which involves learning several similar tasks simultaneously, is able to share the knowledge learned among the tasks.

On the DREAM leaderboard¹, the recent top systems include RoBERTA_{large}+MMM in [Jin *et al.*, 2020] and Albert_{xxlarge}+DUMA in [Zhu *et al.*, 2020]. Both systems employ model architectures composed of a Transformer-based encoder and some matching/attention mechanism between the context and the question-answer pair. RoBERTA_{large}+MMM in [Jin *et al.*, 2020] additionally employed two stages of transfer learning: coarse-tuning with natural language inference (NLI) tasks and multi-task learning with multi-choice reading comprehension tasks.

We used Albert_{xxlarge}+DUMA as our model architecture and did multi-task learning on top of that, as through experiments we felt it efficiently boosts the performance on top of the powerful Albert_{xxlarge} model.

2 Methods

The model architecture is composed of a Transformer-based encoder, a linear layer classifier, and an extra attention layer(s) in between to model reasoning between the context and the question-answer, as in [Zhu *et al.*, 2020]. We use the pretrained Albert_{xxlarge} as the encoder, and fine tune it during the training process. Since the DREAM dataset is small, joint training on both the DREAM task and the RACE task helps get a good boost on the DREAM task.

2.1 Model Architecture

Encoder When encoding an answer option for a question, we concatenate it with not only the question but also the context (passage for the RACE task, dialogue for the DREAM task) to form one single sequence, the parts separated by the `<sep>` token. The sequence is fed through a Transformer-based encoder (Albert in our case). The sequence output of the encoder is then sent to the next part of the model.

Extra Attention layer In the next part of the model, we use Dual Multi-head Co-Attention (DUMA) module as described in Section 4.2 of [Zhu *et al.*, 2020]. Basically, it involves 1) splitting the output sequence from the encoder into two parts, one for the context and one for the question-answer; 2) from the two sequences, computing two attention representations, one from the context attending the question-answer, the other vice versa; 3) the two attention representations are individually mean-pooled and then concatenated together and sent to the next part of the model: the classifier.

Classifier For each question, over all the answer options, the classifier part takes the outputs from the Extra Attention layer, and feeds them through a linear layer. The answer option with the highest probability is picked as the predicted answer. The Cross Entropy function between the ground truth and the predicted probabilities is applied to compute the loss.

2.2 Multi-task Learning

The question/answer pairs in the DREAM task have syntactic and semantic characteristics that are generally different from the text sequences that are used to pre-train Albert. Because the DREAM dataset is relatively small, it is reasonable to hypothesize that adding a larger amount of similar multi-choice MRC data in the training will be beneficial for the DREAM task.

Inspired by [Jin *et al.*, 2020], we did multi-task learning on the DREAM task and the RACE task. Although the number of choices are different in both tasks, we are still able to share all the parts of the model between them. We sampled mini-batches from the DREAM and RACE datasets in proportion of the relative sizes of the two datasets.

3 Experiment Settings

We used the pretrained Albert-xxlarge-v2 model as our encoder, and one layer of DUMA as in [Zhu *et al.*, 2020]. Since we do not have the implementation details on the number of DUMA attention heads and the head size in [Zhu *et al.*, 2020], for re-implementation we used the setting as in Alberta-xxlarge self attention layers: 64 attention heads and each head has a dimension of 64. We used the same setting in our multi-task learning. Our codes are written based on Transformers².

The maximum sequence length was set to 512. We used a mini-batch size of 24, and a learning rate of 1e-05. The gradient norm was clipped to 1. We adopted a linear scheduler with a weight decay of 0.01, trained the model for 5 epochs. For the multi-task learning, we used 10% of the total steps as warming up, evaluated on the dev set at every 1000 steps and saved the best model on the dev set. For the single-task training on the DREAM dataset (the second last line in Table 1), we evaluated on the dev set at every 100 steps and saved the best model on the dev set.

We did not do hyper-parameter searching and only had one run of multi-task learning at the moment. For the single task training we had three runs and picked the model with the best accuracy on the dev set. All the experiments was run on four v100 GPUs in a single machine.

4 Results

Table 1 summarizes the experiment results. The numbers marked with \star are from [Jin *et al.*, 2020], the numbers marked with \dagger are from [Zhu *et al.*, 2020]. Note that the second last line is our implementation of Albert_{xxlarge}+DUMA, with 64 DUMA heads of dimension 64. The multi-task learning in the last line was run with similar settings and parameters.

¹<https://dataset.org/dream/>

²<https://github.com/huggingface/transformers>

model	dev	test
FTLM++[Sun <i>et al.</i> , 2019]	58.1 [*]	58.2 [*]
BERT _{large} [Devlin <i>et al.</i> , 2019]	66.0 [*]	66.8 [*]
XLNet[Yang <i>et al.</i> , 2019b]	[*]	72.0 [*]
RoBERTa _{large} [Liu <i>et al.</i> , 2019]	85.4 [*]	85.0 [*]
RoBERTa _{large} +MMM[Jin <i>et al.</i> , 2020]	88.0 [*]	88.9 [*]
Albert _{xxlarge} [Lan <i>et al.</i> , 2020]	89.2 [†]	88.5 [†]
Albert _{xxlarge} +DUMA[Zhu <i>et al.</i> , 2020]	89.3 [†]	90.4 [†]
Albert _{xxlarge} +DUMA(our implementation)	90.7	88.6
Our model (above model with multi-task learning)	91.9	91.8

Table 1: Results on DREAM dataset, the numbers marked with ^{*} are from [Jin *et al.*, 2020], the numbers marked with [†] are from [Zhu *et al.*, 2020]

Compared to [Zhu *et al.*, 2020], our implementation of Albert_{xxlarge}+DUMA obtained a higher accuracy on the dev set but a lower accuracy on the test set. The possible reasons could be that we did not have the exact setting such as attention head number and size, and/or randomness. Nonetheless, the model from multi-task learning had a good boost over both the numbers from [Zhu *et al.*, 2020] and the numbers from our re-implementation of the single-task learning. This shows that although the context part in the DREAM task is in dialogue style instead of passage style in the RACE task, the DREAM task could still benefit a lot from learning together with the RACE task, because of similar domain and similar question-answer style.

5 Acknowledgement

The author would like to thank Luis Lastras, Sachindra Joshi, and Chulaka Gunasekara for helpful discussions.

References

- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [Jin *et al.*, 2020] Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension, 2020.
- [Lai *et al.*, 2017] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [Lan *et al.*, 2020] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.
- [Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
- [Sun *et al.*, 2019] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 2019.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
- [Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

- [Yang *et al.*, 2019a] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 5754–5764, 2019.
- [Yang *et al.*, 2019b] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019.
- [Zhu *et al.*, 2020] Pengfei Zhu, Hai Zhao, and Xiaoguang Li. Dual multi-head co-attention for multi-choice reading comprehension, 2020.