MY APPROACHES TO SOLVE COLA TASK BASED ON ENSEMBLE MODEL

Zhaoyu Li[*]
Department of Computer Science
Shanghai Jiao Tong University
apollo19990425@sjtu.edu.cn

May 20, 2020

ABSTRACT

The General Language Understanding Evaluation (GLUE) benchmark is a diverse set of existing natural language understanding tasks. In this course project, we choose CoLA as our task to evaluate the performance of our models. For my approaches, I try several models and finally choose two models: ALBERT(from Google Research and the Toyota Technological Institute at Chicago) and ELECTRA(from Google Research/Stanford University) as my pretrained models and then finetune on CoLA tasks respectively. I use ensemble method to combine the results derived from the two models and obtain a quite good score **74.2** on CoLA task (and the screenshot is in the appendix).

1 Introduction

The Corpus of Linguistic Acceptability is a set of English acceptability judgments drawn from books and journal articles on linguistic theory [1]. Each instance is a sentence annotated with whether it conforms to grammar. It consists of 8.5K sentences in train set and 1k sentences in test set. Following the authors, they use Matthews correlation coefficient (Matthews, 1975) as the evaluation metric, which evaluates performance on unbalanced binary classification and ranges from -1 to 1, with 0 being the performance of uninformed guessing. Table 1 shows some examples in CoLA train set.

Table 1: Some exampls in CoLA train set

gj04	1		They made him angry.
gj04	0	*	They caused him to become angry by making him.
gj04	0	*	They caused him to become president by making him.
gj04	0	*	They made him to exhaustion.
gj04	1		They made him into a monster.
gj04	1		The trolley rumbled through the tunnel.
gj04	1		The wagon rumbled down the road.
gj04	1		The bullets whistled past the house.
gj04	1		The knee replacement candidate groaned up the stairs.
gj04	0	*	The car honked down the road.
gj04	0	*	The dog barked out of the room.

2 My approaches

At first, I try ERNIE [2], MT-DNN [3], RoBERTa [4] models with carefully parameters selecting. For ERNIE and MT-DNN, I use the same parameters in the papers but fail to achieve good scores on dev set. For RoBERTa, with

[*] This is a course report for *Natural Language Processing*, CS229, SJTU

the code from https://github.com/pytorch/fairseq, I use the pretrained RoBERTa.large model finetune on the CoLA task. Firstly I choose about 12 different random seeds for the selected hyperparameters and evaluate their performances on dev set. Then I ensemble 7 models' test results based on their dev set metrics and get 67.8 scores on CoLA test set. However, it's hard to improve further. After reading the latest papers and the codebases, I solve CoLA task based on another two models: ALBERT [5] and ELECTRA [6]. Firstly, I use a set of them to finetune on CoLA task and then obtain the predictions respectively, then I use ensemble method to obtain my final result, which get **74.2** score on CoLA test set. The code is available in https://github.com/ApolloLiZhaoyu/CS229_Project.

2.1 ALBERT

Note that although increasing model size improves performance on natural language understanding, further model increases become harder due to memory limitations and longer training times. To address these problems, in paper [5], they present two parameter reduction techniques to lower memory consumption and increase the training speed of BERT as well as using a self-supervised loss that focuses on modeling inter-sentence coherence, and show it consistently helps downstream tasks with multi-sentence inputs. Also in paper [7], they propose a novel adversarial training algorithm, FreeLB, that promotes higher invariance in the embedding space, by adding adversarial perturbations to word embeddings and minimizing the resultant adversarial risk inside different regions around input samples, which improves the performance for natural language understanding.

With the code https://github.com/zhuchen03/FreeLB/tree/master/huggingface-transformers, I use the pretrained ALBERT-xxlarge model with FreeLB algorithm to finetune on the CoLA task. I use the same parameter in the paper [7] and choose about 7 different random seeds. Based on their performance on dev set, I choose top-3 models for prediction on CoLA test set. Their average score on CoLA test set is **69.8**.

2.2 ELECTRA

In paper [6], they propose a more sample-efficient pre-training task called replaced token detection. Instead of masking the input, their approach corrupts it by replacing some tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, they train a discriminative model that predicts whether each token in the corrupted input was replaced by a generator sample or not.

With the code from https://github.com/google-research/electra, I use the pretrained ELECTRA-large model to finetune on the CoLA task. Firstly I choose about 8 different random seeds for the selected hyperparameters and evaluate their performance on dev set. Then I use top-4 models to prediction on test set. All of them reach more than **70.1** score on test set.

3 Evaluation

From my experience to solve this task, I have several findings. Firstly, CoLA task is a small dataset that many models have high-variance performances on it. Consider that, I carefully select hyper parameters, use different models as well as random seeds and ensemble method to solve this issue. Secondly, it's obvious that the pretraining & fintuning deep language model is the mainstream method for natural language understanding and I think it's better to update all the parameters than updating partial parameters during finetuning by experiment results. And I try both cased models and uncased models for CoLA task. It turns out that uncased models perform better, since CoLA task is not case-sensitive. What's more, by reading the latest paper, I think that it's good to use adversarial training algorithm to promote the performances.

4 Conclusion

In this paper, I solved CoLA task based on two models - ALBERT and ELECTRA. I chose different random seeds and combined their predictions on test set, which made me achieve a quite good score. It showed that the latest language models and adversarial training algorithm was powerful for natural language understanding.

References

[1] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[2] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*, 2019.

[3] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[4] 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. *arXiv preprint arXiv:1907.11692*, 2019.

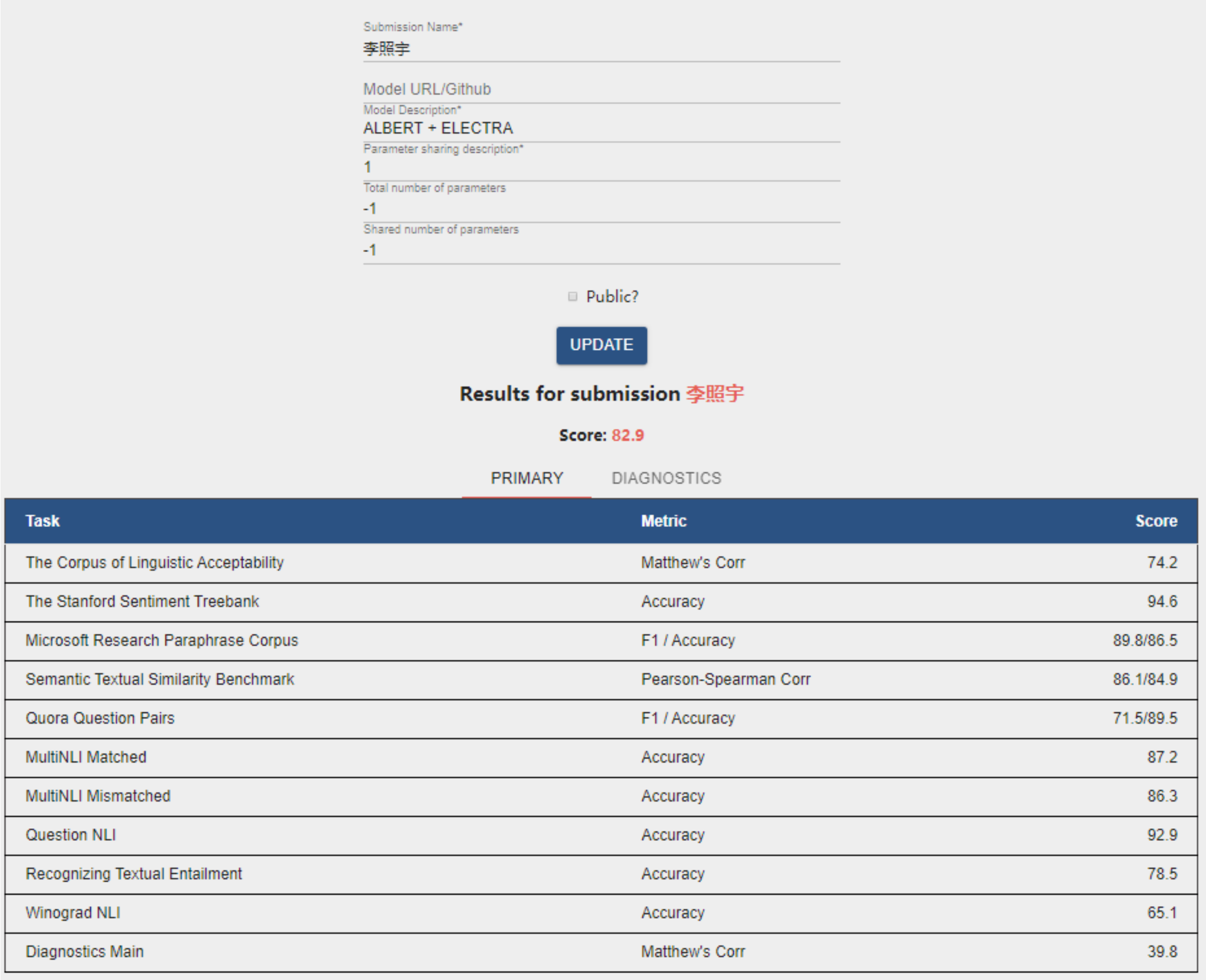
[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[6] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[7] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for language understanding. *arXiv preprint arXiv:1909.11764*, 2019.

Appendix

The screenshot of test result is showed as below:



Submission Name*
李照宇

Model URL/Github

Model Description*
ALBERT + ELECTRA

Parameter sharing description*
1

Total number of parameters
-1

Shared number of parameters
-1

□ Public?

Results for submission 李照宇

Score: 82.9

Task	Metric	Score
The Corpus of Linguistic Acceptability	Matthew's Corr	74.2
The Stanford Sentiment Treebank	Accuracy	94.6
Microsoft Research Paraphrase Corpus	F1 / Accuracy	89.8/86.5
Semantic Textual Similarity Benchmark	Pearson-Spearman Corr	86.1/84.9
Quora Question Pairs	F1 / Accuracy	71.5/89.5
MultiNLI Matched	Accuracy	87.2
MultiNLI Mismatched	Accuracy	86.3
Question NLI	Accuracy	92.9
Recognizing Textual Entailment	Accuracy	78.5
Winograd NLI	Accuracy	65.1
Diagnostics Main	Matthew's Corr	39.8