Natural Language Processing - Individual Final Report

Sisi Zhang

Introduction

The purpose of this project is to build a model that is able to have a good performance on the GLUE(General Language Understanding Evaluation)benchmark. It contains several tasks like a single sentence, similarity and paraphrase and inference tasks which involve 9 different datasets. Our group decided to perform the test mainly using transformers like BERT, XLNet and ELECTRA, then experiment using ensemble model techniques to improve the score. We shared our model prediction result and metric score to find the better parameter and hence improve the score.

Description individual work

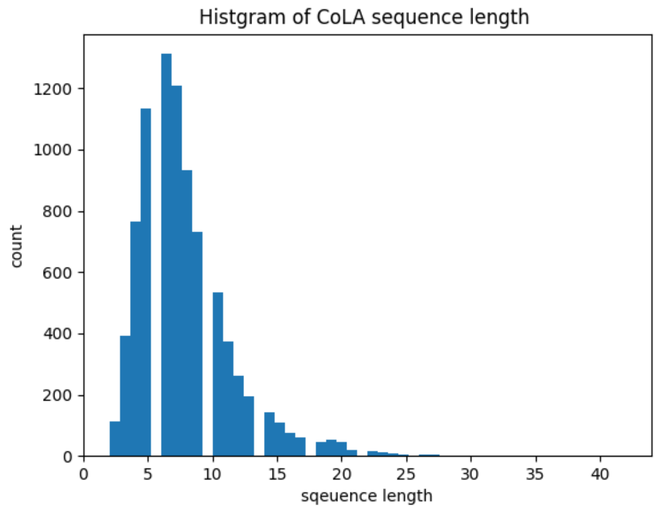
In order to know the glue dataset better. I download the dataset and discover their statistical information. Then I use run_glue.py from the hugging face repository to run different transformers with different datasets. After getting the benchmark, I did search on hyper-parameters and also ensemble modellings.

Describe the individual work in detail

I first download all the datasets and print import information like the head of the dataset, the length of each dataset, and the label value distribution of each dataset to understand the reason for the choice of matrics. Then I use run_glue .py to get the evaluation result of half of the dataset with BERT, XLNet and ELECTRA. And made the table shows all of the results and decide BERT as our benchmark and focus more tunning on ELECTRA.

Task	Metric	Bert-base-cased	Xlnet-base-cased	Electra-base-discriminator
CoLA	Matthews corr	57.01	31.99	65.85
SST-2	Accuracy	92.43	94.15	95.41
MRPC	F1/Accuracy	89.46/84.8	89.76/85.29	90.88/86.76
STS-B	Pearson/Spearman corr.	88.82/88.5	86.73/86.56	90.31/90.35
QQP	Accuracy/F1	90.71/87.49	90.83/87.74	91.79/89.06
MNLI	Matched acc./Mismatched acc.	83.74/84.11	86.38/87.01	88.82/88.67
QNLI	Accuracy	90.32	92.02	92.77
RTE	Accuracy	64.98	64.98	72.56
WNLI	Accuracy	38.03	29.58	35.21

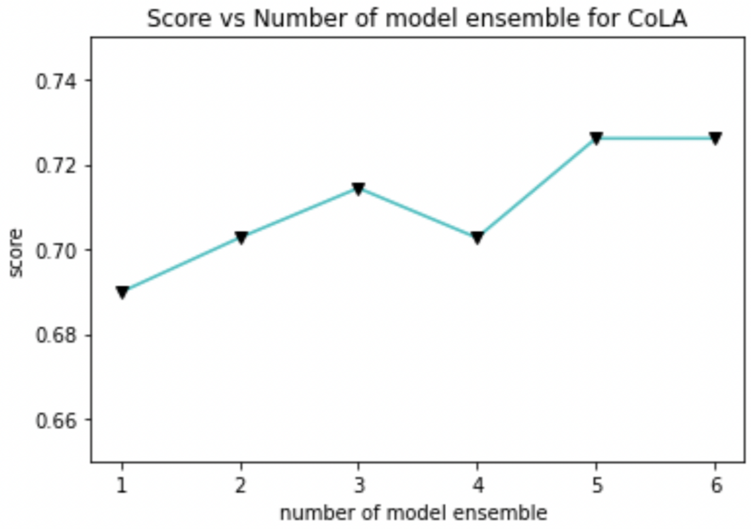
I mainly did hyperparameter tunning search on batch size, maximum sequence length, learning rate and number of epochs. For batch size, I tries 12,20,32,and 64. The result shows when batch size is 32 the model has higher score and less running time. Then I use histogram to decide the porporate maximum sequence length.



Like the graph above, I decide the maximum sequence length as 20 because, at that point, 80 to 90 percent of data is covered. Then I compared the model with the default setting length is 128 and found that they has similar evaluation result but the model with length 20 requires much running time than the other model. For learning rate, I tried 2e-4, 2e-5, 2e-6 and got the result shows that they use similar running time, but the model with 2e-5 learning rate has a higher score. The last one I tried is the number of epochs. I tested with 3, 5 and 10 epoch on the CoLA dataset, they requires 22 mins, 38 mins, 41 mins respactifly and come with a similar score. So I will definitely use epoch 3 for the future models.

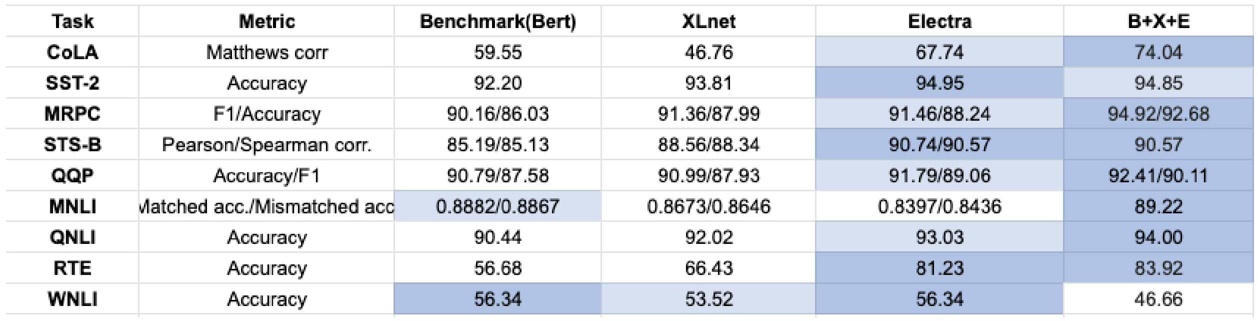
Results

After tunning the parameters, the score increased a little bit. The last thing i try is to ensemble the model from the transformer's prediction to make another prediction. Before we make ensemble models for all the dataset I tasted it on CoLA first. I runed 6 different models to get their prediction on the validation set then combined them with their original labels. After that apply logistic regression on the new dataset to tast out how many models should we ensemble to improve our only one single model left.



The line chart above shows the score changing with the number of model ensembles. As we can see that more model will be more powerful indeed, but due to the time concern, 3 models will be good in our cases.

Summary and conclusions



Task	Metric	Benchmark(Bert)	XLnet	Electra	B+X+E
CoLA	Matthews corr	59.55	46.76	67.74	74.04
SST-2	Accuracy	92.20	93.81	94.95	94.85
MRPC	F1/Accuracy	90.16/86.03	91.36/87.99	91.46/88.24	94.92/92.68
STS-B	Pearson/Spearman corr.	85.19/85.13	88.56/88.34	90.74/90.57	90.57
QQP	Accuracy/F1	90.79/87.58	90.99/87.93	91.79/89.06	92.41/90.11
MNLI	Matched acc./Mismatched acc	0.8882/0.8867	0.8673/0.8646	0.8397/0.8436	89.22
QNLI	Accuracy	90.44	92.02	93.03	94.00
RTE	Accuracy	56.68	66.43	81.23	83.92
WNLI	Accuracy	56.34	53.52	56.34	46.66

This table shows our final result of ensemble model compares to the single models. For most of the task, ensemble models preformem better, but some of them didn't improve that much compare to a single model. So for the future work, I will search on more transformer models and combine them to get a better result. Also for WNLI tast, the ensemble model perform worse than all the single models, I will also do more research on that.

Calculate the percentage of the code
that you found or copied from the internet. For example, if you used 50 lines of code from the internet and then you modified 10 of lines
and added another 15 lines of your own code, the percentage will be 50 − 10 × 100. 50+15

The code I use for discover the statistic information of datasate is my original. The code run_glue.py only add 5 lines code to make it able to print out the prediction csv file. When running the run_glue.py I simplely change the task name and parameters. The code used for testing the number of models to ensemble with preference score is my original.

Reference

Huggingface. (n.d.). Transformers/examples/pytorch/text-classification at master · Huggingface/Transformers. GitHub. Retrieved December 9, 2021, from https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification.

Huggingface. (n.d.). Transformers/examples/pytorch/text-classification at master · Huggingface/Transformers. GitHub. Retrieved December 9, 2021, from https://github.com/huggingface/transformers/tree/master/notebooks.

Clark, M. Luong, Q. V. Le, and C. D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In ICLR.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.