Bradley Reardon
NLP
Final Project – Individual Report

Introduction

The General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems (source). The GLUE benchmark consists of nine natural language processing (NLP) tasks, all of which contain sentences or sentence-pairs used to assess model performance across a wide range of linguistic phenomena found in natural language. A leaderboard exists to compare model performance against fellow contestants across the globe. As a group, we tested various NLP transformer models with varying hyperparameters to test their capabilities and learn to work with the hugging face *transformers* module.

Individual Contribution

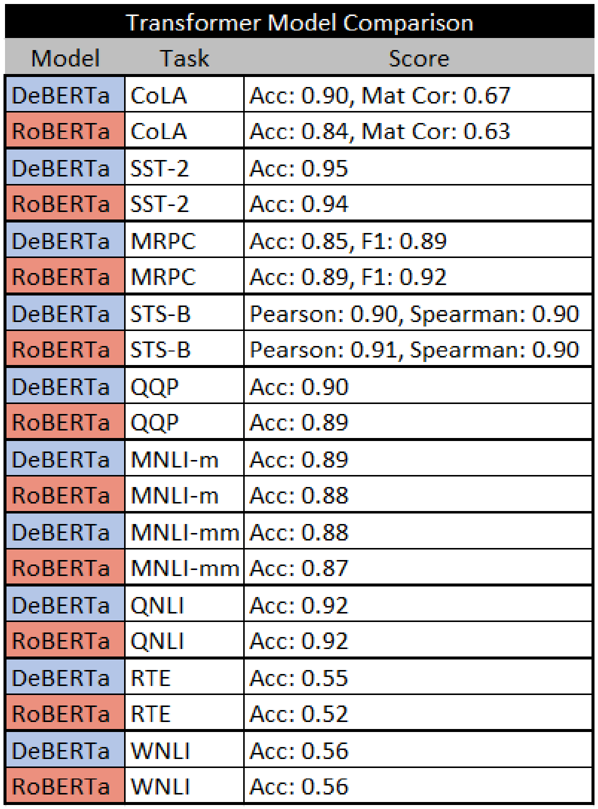
- Code:
  - Scripted the train_MRPC-RTE-WNLI.py file.
  - Tested MRPC, RTE, and WNLI tasks.
- Presentation:
  - Created powerpoint and worked on results slides
  - Presented on results.
- Group Final Report:
  - Intro, results, conclusion

Results

As shown in the following tables, we ran all models using the AdamW optimizer and a learning rate of 0.0005. This was due to time constraints and a lack of processing power allowing us to A/B test all hyperparameters. The number of epochs and batch size varies per task due to dataset size and sequence length. In order to increase efficiency to allow testing of multiple different models, we had to sacrifice performance by using lower epoch counts and the highest batch sizes our cloud hardware would allow. DeBERTa outperformed RoBERTa in COLA, SST-2, STSB, QQP, MNLI matched, MNLI mismatched, and RTE. The models performed equally on QNLI and WNLI, and RoBERTa outperformed DeBERTa on MRPC.

RoBERTa					
Task	Optimizer	Epoch	Batch Size	Learning Rate	Metrics
CoLA	AdamW	4	32	0.00005	Acc: 0.84, Mat Cor: 0.63
SST-2	AdamW	4	64	0.00005	Acc: 0.94
MRPC	AdamW	4	8	0.00005	Acc: 0.89, F1: 0.92
STS-B	AdamW	4	8	0.00005	Pearson: 0.91, Spearman: 0.90
QQP	AdamW	1	8	0.00005	Acc: 0.89
MNLI-m	AdamW	4	64	0.00005	Acc: 0.88
MNLI-mm	AdamW	4	64	0.00005	Acc: 0.87
QNLI	AdamW	4	8	0.00005	Acc: 0.92
RTE	AdamW	4	8	0.00005	Acc: 0.52
WNLI	AdamW	4	8	0.00005	Acc: 0.56

DeBERTa					
Task	Optimizer	Epoch	Batch Size	Learning Rate	Metrics
CoLA	AdamW	4	32	0.00005	Acc: 0.90, Mat Cor: 0.67
SST-2	AdamW	4	64	0.00005	Acc: 0.95
MRPC	AdamW	4	8	0.00005	Acc: 0.85, F1: 0.89
STS-B	AdamW	4	8	0.00005	Pearson: 0.90, Spearman: 0.90
QQP	AdamW	1	8	0.00005	Acc: 0.90
MNLI-m	AdamW	4	64	0.00005	Acc: 0.89
MNLI-mm	AdamW	4	64	0.00005	Acc: 0.88
QNLI	AdamW	4	8	0.00005	Acc: 0.92
RTE	AdamW	4	8	0.00005	Acc: 0.55
WNLI	AdamW	4	8	0.00005	Acc: 0.56



Transformer Model Comparison

Model	Task	Score
DeBERTa	CoLA	Acc: 0.90, Mat Cor: 0.67
RoBERTa	CoLA	Acc: 0.84, Mat Cor: 0.63
DeBERTa	SST-2	Acc: 0.95
RoBERTa	SST-2	Acc: 0.94
DeBERTa	MRPC	Acc: 0.85, F1: 0.89
RoBERTa	MRPC	Acc: 0.89, F1: 0.92
DeBERTa	STS-B	Pearson: 0.90, Spearman: 0.90
RoBERTa	STS-B	Pearson: 0.91, Spearman: 0.90
DeBERTa	QQP	Acc: 0.90
RoBERTa	QQP	Acc: 0.89
DeBERTa	MNLI-m	Acc: 0.89
RoBERTa	MNLI-m	Acc: 0.88
DeBERTa	MNLI-mm	Acc: 0.88
RoBERTa	MNLI-mm	Acc: 0.87
DeBERTa	QNLI	Acc: 0.92
RoBERTa	QNLI	Acc: 0.92
DeBERTa	RTE	Acc: 0.55
RoBERTa	RTE	Acc: 0.52
DeBERTa	WNLI	Acc: 0.56
RoBERTa	WNLI	Acc: 0.56

Summary

When comparing the results of our experiments, we found that DeBERTa overall outperformed RoBERTa, excelling in all tasks other than MRPC. This was expected considering DeBERTa is essentially an improved version of RoBERTa. Our scores on the validation sets ended up being very close to those of the test scores on the GLUE benchmark leaderboard for the majority of the tasks.

In the future, a few changes we could make when experimenting further are increasing the computing processing power and increased variation in our testing. The increased processing power would allow us to experiment with varying hyper parameters such as number of epochs, learning rate, batch size, and optimizer. Due to time constraints and a lack of resources, we were unable to test as many times as we wanted.

If we had not dealt with computing power limitations and time constraints, we would have liked to test out ensembling multiple models. This would most likely increase our metric scores and decrease the gap between our scores and those at the top of the leaderboards.

Percent of Code:

The base code used in my model performance was sourced from the course repository. None of the code in my file was sourced from online.

Citations:

1. https://gluebenchmark.com/
2. https://huggingface.co/docs/transformers/model_doc/deberta
3. https://huggingface.co/docs/transformers/model_doc/roberta

4. https://github.com/huggingface/transformers/blob/5bfcd0485ece086ebcbed2d008813037968a9e58/examples/run_glue.py#L128
5. https://www.kaggle.com/aryansakhala/bert-pytorch-cola-classification#Evaluate-on-Test-Set
6. He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." arXiv preprint arXiv:2006.03654 (2020).