Individual Final Project Report

GLUE TEXT CLASSIFICATION USING XLNET-BASE-CASED

Natural Language Processing (DATS 6312)

December 9, 2021

STEFANI D. GUEVARA

Introduction

The General Language Understanding Evaluation benchmark (GLUE) is a compilation of natural language datasets and tasks designed with the goal of testing models on a variety of different language challenges.

Historically, many NLP models were trained and tested on very specific datasets. This method often produced well performing models when run on similar data but had very poor performance when applied to corpuses that varied from their original training data. Additionally, these models could only be used to perform the specific task they were built for, with very little cross-utility to different tasks, even when using the same data.

With the advent of transfer learning, it became possible to create a model which could be used for multiple tasks, across a variety of input datasets. Through this, models that have a deeper understanding of language can be created, with broad cross-utility and applicability in a variety of natural language contexts.

The 9 tasks in the GLUE dataset represent a diverse set of challenges, from grammatical parsing to context-reliant pronoun identification, from an array of text contexts, including news reports, public forums and Wikipedia pages.

To create a high-performing model capable of understanding and generating natural language, we integrate three state-of-the-art models together with an ensemble learning model to capture information from the three disparate models on each task. With this method, we capitalize on the unique strengths of each individual model, while mitigating most weaknesses.

Shared Work

Each member of the team took on one model available through the Transformers package: Arathi, ELECTRA; Mariko, DeBERTa; me, XLNet. Mariko also took on creating the scripts for the ensemble models. I sourced and edited the script for generating the predictions using the Hugging Face text classification Jupyter notebook under their notebooks repository.
Each member contributed to the preparation of the final report and final presentation, with many thanks for Mariko on her contributions to the report when Arathi and I were limited on time.

Description of Datasets

The GLUE dataset consists of 9 tasks, which can be categorized into 3 broad groups: Single sentence tasks, Similarity and Paraphrase tasks, and Inference tasks. Each of the tasks have discrete datasets, but within groups the datasets may be similar.

Below is a table that summarized the tasks, their respective evaluation metrics, and their sources.

Corpus	Task	Metric	Domain
Single-Sentence Tasks			
CoLA (Corpus of Linguistic Acceptability)	*Grammatical Correctness*	*Matthew's correlation coefficient*	*Linguistics literature*
SST-2 (Stanford Sentiment Treebank)	*Sentiment Analysis*	*Accuracy*	*Movie reviews*
Similarity and Paraphrase Tasks			
MRPC (Microsoft Research Paraphrase Corpus)	*Paraphrase detection of two sentences*	*Accuracy and F1 Score*	*News*
QQP (Quora Question Pairs)	*Paraphrase detection of two questions*	*Accuracy and F1 Score*	*Social QA Questions*
STS-B (Semantic Textual Similarity Benchmark)	*Sentence similarity*	*Pearson and Spearman correlation*	*Misc.*
Inference Tasks			
MNLI (Multi-Genre Natural Language Inference Corpus)	*Sentences match/mismatch*	*Accuracy*	*Misc.*
QNLI (Stanford Question Answering Dataset)	*Question & Answer pairing*	*Accuracy*	*Wikipedia*
RTE (Recognizing Textual Entailment)	*Sentences match/mismatch*	*Accuracy*	*News & Wikipedia*
WNLI (Winograd Schema Challenge)	*Sentences match/mismatch with pronoun substitution*	*Accuracy*	*Fiction books*

Share of Work

I sourced and put together the code used in this project for running the pretrained models from the Hugging Face notebooks repository and included the eval_accumulation_steps parameter to help the models run without hitting “Cuda out of memory”.

I chose to work on the XLNet model, running all tasks and doing the respective research for the project. Additionally, I co-coordinated the project plan and contributed to the Description of the Models, Hyperparameters, Results, and Summary and Conclusion of the paper.

Results

XLNet			
Corpus	**Metric**	**Baseline**	**Tuned**
Single-Sentence Tasks			
CoLA	*Matthew’s correlation*	0.000	0.399
SST-2	*Accuracy*	0.505	0.940
Similarity and Paraphrase Tasks			
MRPC	*Accuracy and F1*	0.686 f1: 0.812	0.892 f1: 0.922
QQP	*Accuracy and F1*	0.372 f1: 0.539	0.874 f1: 0.893
STS-B	*Pearson and Spearman*	P: 0.021 S: 0.010	P: 0.893 S: 0.889
Inference Tasks			
MNLI	*Accuracy*	0.314	0.857
MNLI-MM	*Accuracy*	0.312	0.858
QNLI	*Accuracy*	0.495	0.878
RTE	*Accuracy*	0.459	0.740
WNLI	*Accuracy*	0.578	0.563

The table above shows that while XLNet was able to generate predictions on the baseline -- even fair predictions on the MRPC dataset -- it had a stark improvement when predicting on the validation set with basic fine-tuning. This applies to all datasets save for that of WNLI in which the baseline shows marginally better performance than the fine-tuned model. We can also see from the table that XLNet did particularly well on similarity and paraphrase tasks and remained competitive with its counterparts in this project in datasets within each task category.

Hyperparameter tuning was extremely limited with XLNet since the model, even the base-cased version used in this project, is resource intensive. When attempting to finetune on a given task, the model would crash early on or as much as 2 hours into training. Especially when hyperparameter tuning with Optuna, when training only 5 trials, most tasks would take hours to run all 5 trials and often crash prior to completing. It did not seem worth pursuing hyperparameter tuning extensively since the results that were obtained with Optuna were not competitive to the already achieved scores.

When pitting against the other models in this project, Electra and DeBERTa, XLNet and DeBERTa performed competitively against each other, even with the minimal hyperparameter tuning on each. Out of the 9 tasks plus the MNLI-MM, DeBERTa and XLNet outperformed the other on 5 tasks each. XLNet is not as strongly suited for short sequences given its pretraining on long-range dependencies and masking, in its own way, during permutation; we found this to be a likely reason why the model was beaten in the QQP and QNLI tasks. Questions, inherently, are short sequences where sentences can be much longer.

Summary & Conclusion

Even with little finetuning, XLNet did well including against the other models in this project. Surprisingly, it often outperformed Electra even though it was easier to finetune with Electra. Although XLNet was competitive it was very difficult to run the model on all the tasks, not just because it is computationally expensive – which it undoubtedly is and must be for its permutation strategy – but because conducting even 3 epochs on training for certain datasets took as much if not more than the 12 hours granted by the university's VPN. With a longer project timeframe as well as longer VPN access, it would be interesting to see how much more we could improve the scores of this XLNet-base model on a single GPU.

The code I used came from Hugging Face text classification Jupyter notebook. Only minor edits were made to the code to be able to conduct the experiments, including a line for managing gradient accumulation and a line to remove unnecessary columns from the training set, both of which helped facilitate runtime and eschew "cuda out of memory". Over 90-95% of the code was sourced to allow us to jump-start on working with the models given the short project timeline and the resource intensive training.

References

XLNet: Generalized Autoregressive Pretraining for Language Understanding (GitHub)

Guide to XLNet for Language Understanding

XLNet: Generalized Autoregressive Pretraining for Language Understanding (2019 paper)

Paper Reading #2: XLNet Explained

XLNet Explained (Video)

GLUE Explained: Understanding BERT Through Benchmarks

Dataset description sheet

Glue: A Multi-Task Benchmark And Analysis Platform For Natural Language Understanding (2019 paper)

Hugging Face Text Classification (Jupyter) Notebook

Links to References (in case links do not transfer to PDF)

https://github.com/zihangdai/xlnet

https://analyticsindiamag.com/guide-to-xlnet-for-language-understanding/

https://arxiv.org/pdf/1906.08237v2.pdf

https://researchdatapod.com/paper-reading-xlnet-explained/

https://www.youtube.com/watch?v=naOuE9gLbZo

https://mccormickml.com/2019/11/05/GLUE/

https://docs.google.com/spreadsheets/d/1BrOdjJgky7FfeiwC_VDURZuRPUFUAz_jfczPPT35P00/edit#gid=0

https://openreview.net/pdf?id=rJ4km2R5t7

https://github.com/huggingface/notebooks/blob/master/examples/text_classification.ipynb