DATS 6312: Robert Hilly Individual Report

Introduction

For the final project, myself, Tristin, and Divya worked on the General Language Understanding (GLUE) benchmark. This benchmark contains a set of datasets where the goal is to develop models that display natural language understanding. The kinds of natural language understanding tasks that comprise GLUE range from classifying the sentiment of movie reviews to comparing whether a given pair of sentences have the same underlying meaning.

Each of us contributed to this project by building models for our assigned tasks as well as creating/editing the final report and presentation.

Description of my work

For my part of the project, I fit the *DistilBERT Transformer*, a smaller version of the BERT Transformer, on the following tasks:

- **Quora Question Pairs (QQP)** - A collection of question-pairs where the goal is to create a classifier that correctly predicts whether a given question-pair has the same underlying meaning.
- **Semantic Textual Benchmark (SST-B)** - A collection of sentence-pairs where the goal is to create a model that accuractely models the semantic similarity between two sentences.
- **Microsoft Research Paraphrase Corpus (MRPC)** - A collection of sentence-pairs where the goal is to create a classifier that accurately predicts whether each sentence-pair have the same semantic meaning.

In order to fit DistilBERT to each of these tasks, I pulled the pre-trained model from the HugginßFace Hub, tokenized each dataset with DistilBERT's tokenizer, and then fit the model to each dataset.

The trend in recent years has been to train large language models on massive amounts of data in order to increase their performance on a wide variety of NLP tasks. This trend has proved costly in terms of time, money, and compute power. Moreover, this trend does not amend itself well to students like us who are limited in the aforementioned terms. This is where DistilBERT comes in. In "DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter", Sanh et. al from HuggingFace show that, "[…] it is possible to reduce the size of a BERT model by 40%, while retaining 97% if its language understanding capabilities and being 60% faster" (Sanh et. al 2020). Therefore, with DistilBERT, I'm able to efficiently train a language model using a single GPU.

The DistilBERT architecture is quite similar to the BERT architecture. However, in DistilBERT, "The token-type embeddings and pooler are removed while the number of layers is reduced by a factor of 2" (Sanh et. al 2020).

Detailed description of my work

For my portion of the project, I pre-processed QQP, SST-B, MRPC, and applied DistilBERT to each of these tasks. Here, the HuggingFace Course on Transformers proved invaluable as it helped me better understand:

- The pre-processing steps.
- The HuggingFace API.
- Relevant terminology associated with Transformers.

In terms of writing the report, I:

- Wrote up descriptions of the tasks I worked on.
- Gave an overview of the DistilBERT architecture.
- Contributed my results for the tasks I worked on.
- Edited the entire report in order to make it clear and concise.
- Wrote up definitions for the metrics associated with GLUE.

Results

Below, are the results for the three tasks I worked on:

Table 1: DistilBERT Results on Testing Set

Task	Accuracy	F1 Score	Pearson's R	Spearman
MRPC	85.5%	0.900	NA	NA
STS-B	NA	NA	0.873	0.869
QQP	88.2%	0.843	NA	NA

Based on the results, DistilBERT performs well on MRPC, STS-B, and QQP.

Starting with MRPC, DistilBERT achieved an accuracy of 85.5% and an F1 score of 0.90. Here, accuracy corresponds to the percentage of sentence-pairs that were correctly classified as {having the same semantic similarity, not having the same semantic similarity}, while the F1 score tells us that DistilBERT had a low number of false positives (incorrectly classifying a sentence pair as being semantically the same) and false negatives (incorrectly classifying a sentence pair as not being semantically the same). For MRPC, DistilBERT was trained for 2 epochs with a batch size of 8, a learning rate of 5e-05, and Adam as its optimizer.

Next, we go to STS-B, where DistilBERT got a Pearson coefficient of 0.873 and a Spearman coefficient of 0.869. For the former, this indicates that the predictions DistilBERT made on SST-B have a strong linear association with the target feature in STS-B. For the latter, the predictions from DistilBERT are strongly monotonically associated with the the target feature in STS-B. All in all, these metrics indicate that our predictions are closely aligned to the observed similarity scores in SST-B. For STS-B, DistilBERT was trained for 2 epochs with a batch size of 8, a learning rate of 5e-05, and Adam as its optimizer.

Finally, on QQP, DistilBERT got an accuracy score of 88.2% and an F1 score of 0.843. While this is solid performance, the slightly lower F1 score could indicate that the model has a higher proportion of false positives and false negatives relative to the F1 score observed on MRPC. For QQP, DistilBERT was trained for a single epoch due to GPU constraints, but had the same batch size, learning rate, and optimizer as MRPC and STS-B.

Starting with MRPC, DistilBERT was able to achieve an accuracy of 85.5%, and an F1 score of 0.90, indicating that the model performs well at identifying whether each sentence is a paraphrase of the other. Next, SST-B was achieved a correlation of 87.3 and a Spearman correlation of 86.9, indicating that the predictions of the model line up fairly well with the target feature. Finally, we look at the results for QQP. DistilBERT achieves 88.2% accuracy on the test set and an F1 score of 0.843.

Summary and conclusions

Based on the results from DistilBERT, we can confidently say that it performed well on these set of 3 tasks, even though its architecture was smaller than BERT. Indeed, even with training on smaller batches and not many epochs, we're still seeing very solid performance on each of these tasks.

In terms of what I've learned applying DistilBERT to these tasks, I'd say that even a smaller version of BERT still retains the statistical understanding of the language data it was trained on, which is impressive. Indeed, while larger models do have better performance, smaller models such as DistilBERT opens the door to a wider array of NLP applications for companies that are not the size of Google and Facebook (and for students such as myself who want to experiment with a couple of GPUs!).

Possible improvements would be to increase the number of epochs for these datasets, to see if letting the model train longer leads to a substantive increase in that task's associated metrics. Moreover, it could be interesting to perform some feature engineering on each dataset to see if any handmade features improve the performance of the model.

Percentage of code found/copied from the Internet

About 80-85% of the code was found/copied from the Internet. Again, the HuggingFace course on Transformers proved to be a valuable resource for my work.

References

- "Sanh et. al" "DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter." Mar. 2020. https://arxiv.org/pdf/1910.01108.pdf
- "HuggingFace NLP Course" https://huggingface.co/course/chapter1/1?fw=pt