NLP 244 - Assignment 2 Report

Liren Wu
lwu35@ucsc.edu

1 Introduction

For this assignment, the goal is to compress a fine-tuned transformer based model. Reusing the model and task from assignment 1, when given a utterance, tag the sentence with IOB tags and give the relations. We are given two files: hw1_train.xlsx and hw1_test.xlsx. The train file had the following labels: utterances, IOB Slot tags, and Core Relations. The test file only had utterances. The training data has 2251 examples, and the test data has 981 utterances. An example of the train and test data can be seen in Table 1 and Table 2. We will use supervised learning to help build our models to tag tokens and classify the relation.

2 Models

In this assignment, I was reused my BERT model (Devlin et al., 2019) from assignment 1. A brief overview for what I done last assignment is that I followed and based my code on the section week 3 example. I changed the code to accept the assignment data and made it be able to handle multi-label classification. The original code was able to do the slot tagging and single label classification. I essentially did a one hot encoding for the labels and changed the model to have two different loss functions. I used a CrossEntropyLoss for the slot tagging and a BCEWithLogitsLoss for the intent classifying.

2.1 BERT

I first did some pre-proccesing with the data, the utterances are tokenized by the BERT tokenizer from HuggingFace. The slot labels and intent labels are encoded and all three inputs are put in a dataset and dataloader. Other than than having two different loss functions, I did not change the BERT model from the given started code.

2.2 BERT Quantization

Following the guide from PyTorch, I was easily able to implement dynamic quantization. Surprisingly, it only took about 1-2 lines of code. After fine-tuning the pretrained BERT model, I quantized the BERT model before the evaluation step. Since quantization via PyTorch does not work with the GPU, I had to change the device to CPU after the initial training step.

3 Experiments

I did a 85/15 split for training and validation. From my previous experiments that I did in assignment 1, I stuck with running 3 epochs and a learning rate of 5e-5. From the results I got from assignment 1, it seemed like my model was over-fitting the training set because it performed poorly on the test set and seemed heavily bias toward some labels. To help this issue, I increased the dropout rate from 0.2 to 0.5. My model parameters can be seen in Table 5.

4 Results

Through my testing, quantization seemed to work fairly well. Through dynamic quantization, it was able to reduce the model parameters greatly, while still maintaining close scores. The model metrics and model scores can be seen in Table 3 and Table 4.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

5 Appendix

Utterances	IOB Slot tags	ID
show credits for the godfather	O O O B_movie I_movie	movie.starring.actor
...	...	...
how much did the dark night generate	O O O B_movie I_movie I_movie O	movie.gross_revenue

Table 1: Training data.

Utterances
star of thor
who is in the movie the campaign
...
how many scorsese films were filmed in france

Table 2: Test data.

Models	Sentence Acc.	Slot F1 Score	Intent F1 Score
BERT	91.83%	77.90%	82.57%
Quantized BERT	91.82%	78.61%	78.08%

Table 3: Model Scores (Test set).

Models	Number of Parameters	Size (MB)	Inference Time (sec)
BERT	109,519,921	417	36.20
Quantized BERT	23,874,048	173	23.61

Table 4: Model Metrics Comparison.

Settings	Value
Batch size	1
Epochs	3
Learning Rate	5e-5
Dropout	0.5
Optimizer	AdamW
Loss function	CrossEntropyLoss and BCEWithLogitsLoss

Table 5: Best Configuration.