Fine-tuning Contextual Language Models for Language Understanding Tasks
NLP 244, Advanced Machine Learning for NLP Homework 1

Instructor: Dilek Hakkani-Tur
TA: Jonathan Scott

Due: April 25, 2021 5:00PM

1 Introduction

The goal of this homework is to fine-tune transformer based machine learning models to determine knowledge graph relations (in this case, according to the Freebase schema) that are invoked in user utterances to a conversational system, as well as tag words associated with slots. You will be given a training set of utterances paired with a set of IOB slot tags and relations, that you can use to train models to predict the corresponding relations and slots for a given utterance. Here is an example utterance from the dataset:

`Show me movies directed by Woody Allen recently.`

There are two relations that are invoked by this utterance:

1. `movie.directed_by`
2. `movie.initial_release_date`

And here is the sequence of slot tags, associated with each of the input words: `O O O O O B_director I_director` The goal is to use the *HuggingFace* library to train transformer models and output the associated set of IOB tags and relations when given a new utterance. We recommend smaller transformer based models due to computation constraints. Here are some example models you can try:

- `DistilBERT`
- `ALBERT`

Try to implement and experiment with at least two of these models, so that you can compare the performance of various models to build a better intuition of their suitability. Test out key hyper-parameters and find what works best.

One more aspect to consider is that this homework requires you to do multi-label classification. You may reach a certain level of accuracy by predicting only a single class label, but to reach higher values (and potentially 100%), you will need to predict more than one class. Refer to *Scikit-Learn* documentation[1] for more information about multi-label prediction.

2 Dataset Format

This dataset is generated based on film schema of Freebase knowledge graph. There are four files.

2.1 hw1_train.csv

The training data set has three columns:

1. **ID:** the example id for each row.
2. **UTTERANCE:** the natural language text you will need to extract relations from
3. **IOB Slot tags:** the IOB slot tags for each token in the utterance.
4. **CORE RELATIONS:** the relations invoked in the utterance.

2.2 hw1_test.csv

The test data. It has two columns:

1. **ID:** the example id for each row. *Note: Please do not mix it with the IDs in training set.*
2. **UTTERANCE:** the text you will need to predict the relations and slot tags for it.

2.3 sample_submission.txt

A sample submission file. You will need to submit your predictions in the same format. It is a tab separated text file with two columns:

1. **IOB Slot tags :** the slots predictions associated with utterances in *hw1_test.csv*.
2. **CORE RELATIONS:** the relations you predicted from the corresponding utterances in *hw1_test.csv*.

[1] `https://scikit-learn.org/stable/modules/multiclass.html`

2.4 evaluation.py

A script you can run in the terminal using `python3 evaluation.py`, you should store your submissions as shown in the `sample_submission.txt` but name the file `prediction_dev.txt`. The script will return sentence level accuracy, slot F1 scores and intent F1 scores.

Please make sure these are in the same order as the input test file, and print out an empty line for those examples where your code did not predict any output. Hence, there should be the same number of lines in your submission file as the input test file.

3 Submission Files

You need to submit your predicted IOB Slot tags and relations on Canvas, see Section 3.3. The submission must be a txt file named `prediction.txt` uploaded in a zipped format. Besides, you are also required to submit one report as well as your training and inference code on Canvas. The code package should include everything that would be required to train and use your models at run-time. Please ensure you set up your code accordingly for us to be able to replicate your training and submissions — this involves setting a random seed using `numpy.random.seed` or `random.seed`. You need to describe your models, approach, and parameters in your report.

4 Evaluation

Submissions will be evaluated based on their mean *F-1 score* for slots and relations. The *F-score*, commonly used in information retrieval, measures accuracy using the *Precision* **P** and *Recall* **R**. *Precision* is the ratio of true positives (**TP**) to all predicted positives (**TP + FP**). *Recall* is the ratio of true positives to all actual positives (**TP + FN**). The *F-1* score is given by

$$F_1 = \frac{2PR}{P+R}$$
$$P = \frac{TP}{TP+FP}$$
$$R = \frac{TP}{TP+FN}$$

Note that the F-1 score weights recall higher than precision. The mean F-1 score is formed by averaging the individual F-1 scores for each row in the test set.

We also evaluate based on the sentence level *accuracy*, which measures exact matches of your predictions with the true values.

5 Homework Report

The homework report must be a detailed summary of the approach you followed to complete your work. We highly recommend that you use a LaTeX template[2] for your report since for your proposal and final project, you will need to prepare those using the *ACL Proceedings* format. You will be required to provide the following high level sections as part of your report with additional subsections as described:

5.1 Introduction

1. Provide a formal statement of the problem you are trying to solve — whether it is a supervised or unsupervised problem, what specific task it is.

2. Describe the dataset that was provided to you — background information, descriptive statistics of the dataset, what the input and output of the dataset are. Provide examples from the dataset inline or in tables.

5.2 Models

1. Give a description of what features have used for the models that you are training and how these features are created from the input data. Give an illustrative example of the input and output.

2. Include a subsection for each model that you are training. Give a brief summary of how the models are implemented, trained, and the tuneable hyperparameters of the model. Additionally, provide citations to the original work that implemented these methods.

5.3 Experiments

1. Provide a description of the data-set split, the method for selecting hyper-parameters of your final models, any approaches you used for handling data sparsity/imbalance.

2. Include a subsection for each model to describe the different values for the hyper-parameters tested, any special configurations for your model such as solvers or algorithms used.

3. Describe the methods you used to evaluate how good your models were and what criterion you used to select the models for generating your test set submissions.

[2] `https://www.overleaf.com/latex/templates`

5.4 Results

1. Describe how well each model performed on your train, validation and test sets. Describe how the performance varied with different choice of hyperparameter values. Include the requisite tables, plots and figures to illustrate your point.

2. Identify the best performing approach(es) and validate why they performed well. Try to bolster your conclusions by finding and citing work which arrive at similar results.

5.5 References

Provide a bibliography for the literature that you cited. You can make use of `bibtex` or `natbib` packages to automatically generate the bibliography section.

5.6 Appendix

Include an appendix for more detailed table, plots and figures, if needed.

6 Timeline

1. Canvas report and code deadline: 04/25/2021 5:00 PM