NLU project exercise lab: 10

Mateo Rodriguez (239076)

University of Trento

md.rodriguezromero@studenti.unitn.it

The objective of the lab was to compare the performance of 4 different models, some based on LSTMs and one based on BERT. The dataset used was the ATIS dataset which contains labeled slots and intents and the models were trained to predict both at the same time. Slot prediction performance was the highest with LSTMs while Bert handled intent prediction better.

1. Introduction

- For the first part of the lab, the three models used were: a base model (ModelIAS) composed of an LSTM followed by two fully connected layers, the same model but now with a bidirectional LSTM, and the last modification was adding dropout to this bidirectional model. The task in hand is slot and intent classification using the ATIS dataset.
- For the second part a pre-trained BERT model was used along side hidden layers to predict both slots and intents at the same time. The tokenizer was capped at 128 tokens while the utterance in the train data with the most amounts of tokens was only of length 52

To maintain the variables as similar as possible the loss used for all the models was the same: a sum of the cross entropy for intents and for slots.

2. Implementation details

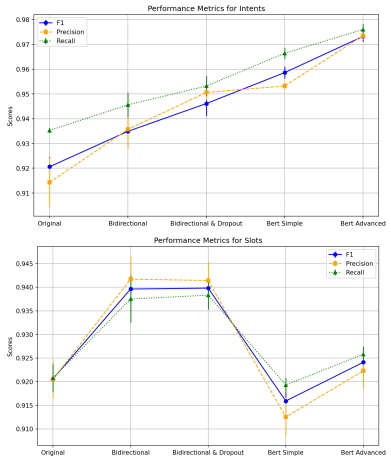
The 4 models implemented in both parts of the lab are relatively straight forward. The original model (and the 2 variations tested) are based on LSTMs to process the text which has been already outperformed in various NLP tasks by transformer architectures like BERT. Nonetheless, LSTMs are still capable of solving some tasks with ease, and this is what was observed during the lab. The models that use LSTMs got progressively better with each modification, and their results are just slightly inferior to the results obtained with BERT. All models were trained using Adam and a batch size of 128. Regarding evaluation it was decided to use F1, precision and recall since they evaluate performance in different ways, taking into account the ability to identify positive instances, avoid false positives, and handle imbalances in the dataset.

To be able to make the BERT model train for two tasks at once it was necessary to use two different linear layers following the outputs of BERT. The first linear layer takes as input the last hidden state of BERT's CLS and is used to obtain predictions for slot labels. The second linear layer is uses instead the whole last hidden layer. This basic modification on top of BERT yielded good results, it outperformed the other models in the intents, but struggled with the slots.

To increase the slots results it was necessary to add some extra modifications to the branch that was in charge of that task. First a dropout layer of 0.1 was added to the input of both linear layers, this boosted results but still didn't surpass the best LSTM. From here I added another linear layer and another dropout this time only to the processing of the slots in between each linear layer. This boosted results marginally, still not being able to beat the LSTM results. The final configuration I ended up with was 1 linear layer for the intents predictions and 3 linear layers for the slots prediction, these layers I added batch normalization and Mish as an activation function.

3. Results

As mentioned before the metrics used to evaluate the models were precision, recall and F1 score. The following figure shows the scores of each of the model variations tested. For the BERT models both results were included, the simpler model with a single linear layer for both intents and slot prediction, and the more complex one described previously.



As evidenced in the figures, as the complexity of the model increases the metrics for the intent predictions increase almost linearly, which is to be expected. What's surprising is the lackluster performance of the BERT models regarding slot predictions compared to the simpler LSTMs models, specially considering the 'advanced' configuration which uses more memory. This results may be due to a data imbalance or the dataset being too small for a complex model as BERT leading to problems generalizing, in fact the validation F1 scores were around 0.97 while the test scores were close to 0.92 further hinting that this may be the cause of the problem.

Other possible causes could be a limitation of the representation obtained by BERT for the specific task of predicting slots, which may be originated in the wordpiece tokenizer making it difficult to relate sub-word with complete slot labels.

In conclusion the tests show that despite LSTMs being mostly replace by transformers they are still relevant for simpler tasks and can perform quite well.