NLU course project - Lab5 - Part2

Emanuele Poiana (247176)

University of Trento

emanuele.poiana@studenti.unitn.it

1. Introduction

Project part two posed two tasks: slot filling and intent classification. Like part one, in the A section one model is proposed and need to be tuned and expanded, then in the B section we fine-tune on the same dataset a pretrained BERT to do the tasks and evaluate through accuracy(intent) and f1 score(slot).

Tuning the architecture hyperparameters was the first step also in this part, so I conducted different tests to pick determined ranges of values. Then I applied sequentially bidirectionality to the IAS model LSTM, and add two dropout layers for regularization. For the fine tuning part I first took and tested the pretrained BERT from HuggingFace. to understand the tensor manipulations. Secondly following the [1] implementation I extend the model with two separate layers and managed the sub-tokenization mismatch for the slot filling task.

Finally, I implemented the methods for train and evaluate the extended model and applied them to train and confront with the best IAS modified model.

2. Implementation details

1. Baseline setup: try different combinations of hyper-parameters to find the best one
2. Bidirectional LSTM: switch the LSTM layer with a bidirectional one (torch implementation)
3. Dropout: add a dropout layer inside the model and tune the new hyper-parameters
4. BERT Fine-Tuning: Create a new class which represents the extension of BERT and implement the train and evaluation loops for it.

I set up a baseline IAS model, taken from Lab 5, using accuracy for the intent classification and f1 score for the slot filling as measures for comparison. I added a Bidirectional LSTM layer instead of the unidirectional one, this required also to modify the Linear output layers to be adapted to the new size of each sequence element, now doubled in its hidden dimension. To compute the output intent logits I used only the reverse hidden dimension logits of the **first** element of the sequence, this to ensure to get the hidden state with all the sequence encoded. For the output slot logits I used the entire sequence.

Then I introduced a dropout layer before the passing the hidden states to the output layers, in this setting both the Linear output layers will received the same masked input.

Finally, I expanded the pretrained BERT model with two output layers, that outputs for each sample of a batch the maximum sequence length of that batch. Consequently special tokens have to be handled and also the sub-tokenization of BERT. I implement a collate function that loads the samples, PAD them and assign a PAD value (0) to the sub-tokens and updates the target slots with this values. Thanks to this is possible to remove or ignore after the computation al the tokens that are not relevant for the loss computation or the performance evaluation.

The train and evaluation loops are conceptually identical to the Lab 5 implementations

3. Results

3.1. Setup

All the part_A experiments ran up to 200 epochs with gradient clipping = 5 and patience = 5, due to the high epochs number, a logging interval of 5 had been chosen number of runs to average 10. AdamW was the optimizer with lr = 0.0001 and weight decay = 0. The IAS model had embedding size of 300 and output size of 400, these parameters have been found through testing but the difference was really minor, see 1. For the Bidirectional models, I halve the hidden size to maintain the same dimension as the IAS model.

3.2. Discussion

IAS: Due to the small dataset, already the starting model set a high standard, probably overfitting.
Bidirectional: The bidirectional IAS model encode better the context of the sequence and achieves better results
Dropout: The dropout in this situation given the already high scoring baseline probably doesn't enhance the training but instead mask logits which represent important information.
BERT: Definitely this pretrained model represent a powerful encoder, that on such small dataset and high baseline easily match and overcome the part A models. On the negative side, this is on order of magnitude bigger in memory terms and probably more in terms of computational resources.

4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," 2019. [Online]. Available: https://arxiv.org/abs/1902.10909

Model	Accuracy	f1 Score
IAS	0.94	0.92
Bidirectional	0.94	0.94
Dropout	0.94	0.93
BERT	**0.97**	**0.96**

Table 1: *Best Accuracy and F1 scores for all the models.*

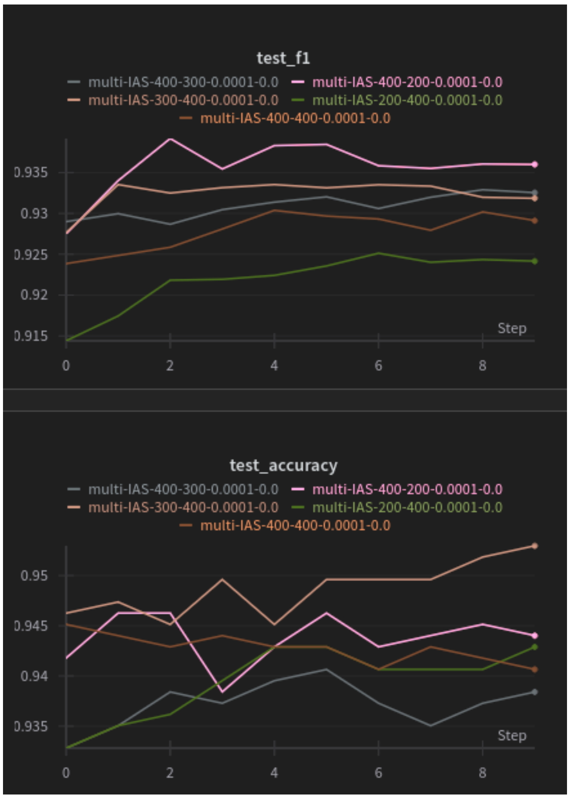

Figure 1: *Part 2A, comparison of baseline IAS model trained with different hidden and embedding dimensions size, the difference is in the range of 2 percentage points*