NLU project exercise lab: 10

Matteo Minardi (238789)

University of Trento

matteo.minardi@studenti.unitn.it

1. Introduction

In the first part, the goal was to modify the baseline ”ModelIAS” to add a set of improvements and see how these affect the performance. I had to play with the hyperparameters to maximize Slot Filling F1 and Intent accuracy. In particular:

- Add bidirectionality
- Add dropout layer

using the ATIS dataset for evaluation.
The second part was about the fine-tune of a BERT model using a multi-task learning setting on intent classification and slot filling. All the necessary information to implement anything was taken from the papers cited in the references.

2. Implementation details

The best results were when the implementations were added one on top of the other in cascade.
The first thing that was needed to be done is the mapping of the slots and the intents of the dataset onto numerical ids so that they could be used to evaluate the model.
Then, after preparing the datasets and the dataloaders, the bidirectionality was implemented doubling the hidden size for the output of the slot filling, since now the information has in to flow in both ways and the predictions depend on both the past tokens and the future ones. By processing data in both directions, the model is able to better understand the relationship between sequences. The hidden size of the intent classification is returned as it is because the number of sentences remains the same.
The dropout layer was implemented so that it is only introduced when explicitly requested by the boolean flag. The dropout probability was kept at 0.1 because it was the one which provided the best results during some empirical testing.
Compared to the baseline, both approaches provided an improvement in the results.They were tested on 5 runs of 200 epochs each, with a patience of 3.

In the second task, before feeding the data to the classifier, in the collate function I decided to pad all sequences of the dataset with the same length, so that there wouldn't be any differences between batches. The selected length is the maximum length of a sentence found in the corpus right before the explicit declaration of the train, val and test datasets.
After this step, it was possible to feed BERT in order to obtain the last hidden state's output for slot filling and the pooler output for intent classification. These values are sent to their respective finetune-able linear layers and are also used in dropouts before being sent to compute the loss.

3. Results

Model	Slot F1	F1 sd	Intent Acc	Acc sd
ModelIAS	0.917	0.005	0.934	0.005
Bidirectional	0.94	0.004	0.948	0.004
Bidirectional + dropout	0.94	0.003	0.95	0.003

Table 1: *Best configs of part 1 (sd = standard deviation)*
The increase in performance is very significative when adding the bidirectionality, increasing both the slot filling F1 and the intent accuracy. Even adding the dropout improved the results, making also the predictions less variable and reducing the uncertainty.

Model	Slot F1	F1 sd	Intent Acc	Acc sd
Fine tuned BERT	0.913	0.002	0.976	0.002

Table 2: *Best configs of part 2 (sd = standard deviation)*
Compared to the first part, the fine tuned BERT has little lower F1 than the LSTM improvements with just bidirectionality or with also dropout, but was able to achieve a better accuracy on the Intent classification, also with smaller standard deviation.

4. References

[1] Schuster, Paliwal. 'Bidirectional Recurrent Neural Networks' 1997. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=650093

[2] Chen, Qian, Zhu Zhuo, and Wen Wang. 'BERT for Joint Intent Classification and Slot Filling' 2019. http://arxiv.org/abs/1902.10909.

[3] HuggingFace community. 'Summary of the tokenizers'. https://huggingface.co/docs/transformers/tokenizer_summary