05 - NLU

Marina Segala (mat. 248447)

University of Trento

marina.segala@studenti.unitn.it

1. Introduction

In the exercises of the NLU laboratory, the goal was to evaluate the performance of two tasks: slot filling and intent classification. The first one is evaluated thanks to the usage of the *conll script*, and generally the F1 score is considered while for the second one, the *classification_report* is taken into account, benefiting the calculus of the accuracy.

By adjusting the values of hyper-parameters present in the model, I made several little adjustments and improvements: in this way I provided more consistency and avoided significant discrepancies, also integrating the state-of-the-art and results of the previous steps to optimize the final results. [1]

2. Implementation details

Starting from an LSTM as the model architecture, as an initial improvement, I decided to make the model bidirectional. In this way an additional layer is added for reversing the direction of information flow. To permit that, the size of each input size that passes into the slot_out and intent_out layers has to double. Moreover, in the forward process implemented, I had to modify also the procedure for extracting the last hidden layer, due to the addition of the bidirectionality.

Seeing the obtained result, another improvement that was possible to make was the insertion of the dropout layer, applied to the last hidden state and to the unpacked output of the model.

After this first implementation, I decided to apply the pre-trained BERT model to get the results of the intent classification task and slot filling one. [1]

In this case, starting from the 'backbone' used previously, I had to do some adjustments to classes and functions. After the importation of the tokenizer, the config file and the model of *'bert_base_uncased'*, I modified the class used for the creation of the train, evaluation and test datasets: also the inputs_ids and the attention_mask had to take into account and I had to pay attention to the creation slot_id too. In fact the tokenizer of Bert creates a list of tokens, containing also two special tokens for indicating the beginning and the end of the phase (in this case 'utterance'), in which some words can be treated with the the subword tokenization strategy. Therefore every time I saw a token equal to '[CLS]', '[SEP]' or if it starts with '##' (subword tokenization) I had to insert into the list of slots a 0 [Table 3]. As a consequence the attention mask has been filled with 0 when the corresponding token is 'pad' (and its slot is equal to 0) and with 1 otherwise.

Before using the model I decided to initialize the weights of those linear layers that I added - *slot_out* and *intent_out*.

Then the training and evaluation phases started. The majority of changes were done in the 'eval_loop', in particular in the 'slot inference step': to understand and be able to confront the list of the predicted slots with the real one, I had to remove the slots in which the value is 0 (or respectively the id is equal to 'pad'), which correspond to a sub-tokenization word or a special token.

3. Results

Incrementally steps and adjustments have permitted to increase each time the performance of the model, also thanks to a try-and-error approach. To have an idea of the behavior of the model I decided to plot data: for the first part, I plot together the F1 score and accuracy obtained in the evaluation steps done with the test dataset for each run. While to see the functioning of the model implemented using a pre-trained version of Bert, I plotted the loss. In the end, the final version of the model with pre-trained Bert works better than the one with LSTM.

Steps	F1 score	Accuracy	Uncertainty
Bidirectionality	0.937	0.941	0.002
Dropout	0.938	0.948	0.002

Table 1: *First part of implementation techniques*

Adjustments	F1 score	Accuracy
Batch size $= 128$ and lr $= 1e{-6}$	0.929	0.956
Batch size $= 64$ and lr $= 1e{-6}$	0.932	0.969
Batch size $= 64$ and lr $= 1e{-4}$	0.950	0.972

Table 2: *Code implemented with Bert*

4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," 2019.

Utterance		I	'd	like	to	find	the	cheapest		fare	
Slots		O	O	O	O	O	O	B-cost_relative		O	
Tokens	CLS	I	d	like	to	find	the	cheap	##est	fare	SEP
Slots_id	0	44	44	44	44	44	44	69	0	44	0

Table 3: *Example of the tokenization and how adjust labels*