NLU project exercise lab: 10

Giovanni Scialla (239181)

University of Trento
giovanni.scialla@studenti.unitn.it

1. Introduction

The aim of this work is to train a multi-task language model based on LSTM Recurrent network (Part 1) and BERT [1] (part 2). In particular, both models had to predict **Intent classification** and **Slot filling** on the famous aviary **'ATIS'** dataset In the first part, a bidirectionality LSTM has been used with the adding of some dropout layer while in the second part we substitute the LSTM layer with the uncased version of BERT base from the notorious transformers library 'HuggingFace'[2]. The BERT version has shown a major improvement in the intent classification task and training time, while dropping a bit on the slots predictions.

2. Implementation details

In order to evaluate both the slot filling and intent classification tasks i have to, firstly extract all the possible slots and intents across the whole dataset, and then build a language mapping from text to token IDs.
For the first part of the laboratory, the bidirectionality has been added making sure to double the input size of the last linear layer since now the LSTM network will return both the first and last hidden state, which is a fair assumption if you consider that the "meaning" of a text comes from both the following and previous words. The output of the recurrent network is then passed through 2 different linear layers, containing as output size the number of slots and intents classes respectively. In the BERT version instead, before feeding the data to the classifier, an extra step is needed in order to find the best maximum length for the bert's tokenizer. to find the maximum length parameter i've plotted an histogram of tokenized sequencies density as shown in figure 1. With a maximum length of 52 we can tokenize the whole dataset by adding ¡PAD¿ tokens. Otherwise, to speed a little the training process, it is also possible to use as maximum sequence length 32 and still include the vast majority of the dataset. After the tokenization is time for BERT to shine, its output contains the last hidden state which has a representation for each single word of the utterance. we can use that for the slot filling task while the pooler output contains the representation for the whole sentence that is used for the multi class intent matching. Both the bert's outputs are then passed through their respective linear layer for fine-tuning.

3. Results

Both the models have been evaluated using the F1-score for the slot filling task and the accuracy score for the intent classification task.
To assess a good evaluation of the models, all of them have been trained from the initial state for 5 runs. The best model is the one that had the higher sum between intent accuracy and f1 slot score along all of the runs.

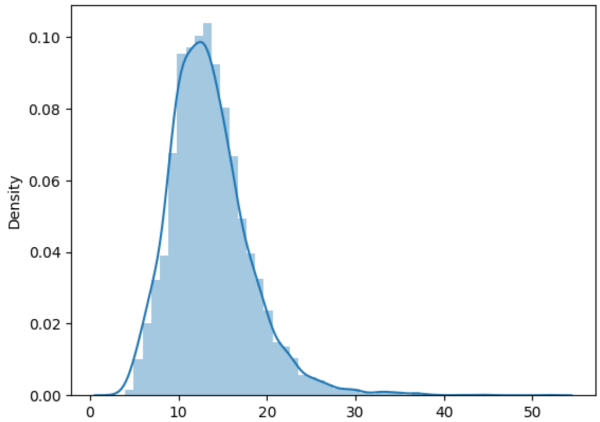

Figure 1: *Density of the Bert's tokenizer sequence length*

	Slot Filling	Intent Classification
LSTM (bidirectional)	0.942	0.955
BERT (fine-tuned)	0.915	0.976

4. References

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

[2] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," 2020.