NLU project exercise lab: 10

Massimo Stefan (240998)

University of Trento

massimo.stefan@studenti.unitn.it

1. Introduction

I embarked on a two-step enhancement of the Model IAS for intent classification and slot filling. In the first step, I modified the baseline architecture by introducing bidirectionality to capture dependencies in both forward and backward directions. Additionally, I incorporated a dropout layer to prevent overfitting and improve the model's generalization capabilities.

In the second step, I leveraged the BERT model, a state-of-the-art transformer architecture, to train a multi-task learning system. This system was specifically tailored to simultaneously handle intent classification and slot filling tasks. I aimed to create a robust and efficient model capable of understanding and processing natural language in a more nuanced manner. The implementation was aligned with the exercise's objectives and was designed to optimize performance on both tasks.

2. Implementation details

In the initial phase of the project, I focused on enhancing the baseline Model IAS. The baseline Model IAS was modified to include bidirectionality and a dropout layer. The bidirectional LSTM ('nn.LSTM') allowed the model to process sequences from both directions, capturing richer contextual information. The input size for both the slot and intent output layers was adjusted to accommodate bidirectionality (hid_size * 2), and the dropout layer ('nn.Dropout') was added to reduce overfitting.

In the second phase of the project, the emphasis was on building a comprehensive pipeline for slot-filling and intent classification using a pre-trained BERT model. The data was divided into training, development, and testing sets, with a stratified split based on intents to ensure that even infrequently occurring intents were properly represented.

The Lang class was created to handle slot and intent vocabularies, including the special padding token. Special tokens [CLS] and [SEP] were added to the tokenized input sequences, and a careful alignment procedure ensured that the slot labels corresponded with the tokenized words, even when words were broken down into sub-tokens.

The model itself was built around the pre-trained BERT model, adding dropout layers for regularization and separate classifiers for slot filling and intent classification. The data was then packed into mini-batches using a custom collate function, ensuring that sequences were properly padded, and, if necessary, efficiently packed for RNN processing.

The training process involved optimizing the model using the AdamW optimizer and training both the slot-filling and intent classification tasks simultaneously. Cross-entropy loss was used for both tasks, with the padding token being ignored in the slot-filling task.

3. Results

The training loop (train_loop) included the calculation of both slot and intent losses using Cross-Entropy Loss ('nn.CrossEntropyLoss'), with the sum of these losses used for backpropagation. The gradient was optionally clipped to avoid exploding gradients. The evaluation loop (eval_loop) was designed to compute the loss and make predictions for both slots and intents.

The train_and_eval function wrapped the training and evaluation loops, implementing early stopping based on the F1 score for slots. The training and evaluation were conducted for 100 epochs with a patience of 3.

The results from both parts of the assignment demonstrated the effectiveness of the implemented models:

- **Part 1**: the enhanced baseline Model IAS, which included bidirectionality and a dropout layer, achieved a Slot F1 score of 0.9631 and an Intent Accuracy of 0.9798;
- **Part 2**: the BERT-based model for intent and slot filling exhibited even better performance, with a Slot F1 score of 0.9805 and an Intent Accuracy of 0.9849.

The improvements seen in Part 2 can be attributed to the utilization of the BERT model, which has been pre-trained on a large corpus, thereby capturing more generalized and rich semantic information. The careful alignment of tokens and labels, along with the simultaneous training for slot-filling and intent classification, contributed to the effectiveness of the model. Overall, these results validate the chosen modeling approaches and underscore the potential of deep learning methods in natural language understanding tasks.

Scores	Values
Slot F1	0.9631
Intent Accuracy	0.9798

Table 1: *ModelIAS performances*

Scores	Values
Slot F1	0.9805
Intent Accuracy	0.9849

Table 2: *BERT performances*

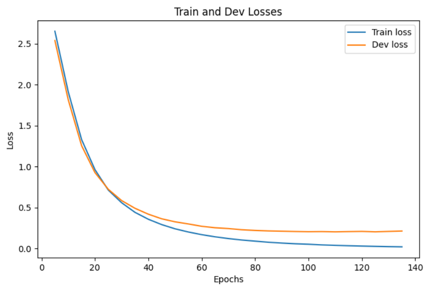

Figure 1: *Losses of ModelIAS*