NLU second course project

Carlotta Giacchetta (248321)

University of Trento

carlotta.giacchetta@studenti.unitn.it

1. Introduction

This report illustrates the development of a neural network for slot filling and intent classification.

In the first part of the assignment, an LSTM network is modified to perform these functions jointly, introducing a bidirectional architecture and a dropout layer to increase generalization and efficiency.

Then, in the second part of the assignment, the focus shifts to a BERT model, which is refined to improve classification capabilities. This phase involves the adaptation of the pre-trained model for a more precise analysis and interpretation of the context and meaning of user input, with the objective to increase the performance of the model in respect to the classical LSTM.

2. Implementation details

2.1. First phase

Throughout the project, significant improvements were made to the pipeline to boost the performance of the intent and slot filling model. Initially, I expanded the LSTM network with a bidirectional architecture to better capture contextual information from the input sequence. I then added two dropout layers to the updated LSTM model. The first was positioned after the embedding layer to mitigate overfitting by randomly deactivating neurons during training, and the second was placed before the output layer to further strengthen the model's robustness against overfitting.

2.2. Second phase

In the second phase, I refined the BERT model, starting with importing the pre-trained model and tokenizer from Hugging Face. I established a fine-tuning pipeline, using BERT's tokenizer to process utterances, keeping only the first token of multi-token words. I created a custom dictionary for slots and utterances, assigning numeric tokens by simple counting. The data loader was configured to handle the attention mask during batch assembly. For fine-tuning, I introduced a class incorporating a dropout layer and two linear layers—one for intent classification and one for slot filling. During the forward pass, I applied dropout to the last hidden state and pooler output, then directed them to the linear layers. I also adjusted the token-to-word conversion method during the eval loop, using BERT's tokenizer for utterances. [1]

2.2.1. Others improvements

I developed a new tokenization strategy for utterances, keeping all generated tokens instead of following the reference paper's method. This caused a size mismatch between the ground truth and the input. To address this, I implemented a function that adjusts the ground truth to match the tokenized utterances, ensuring proper alignment for training.

3. Results

To evaluate the effectiveness of the implemented models, I fine-tuned the parameters in both phases of the project.

3.1. First phase

During the first phase, I primarily focused on optimizing the learning rate (lr = [0.0001, 0.01, 1]), dropout probability (dp = [0.1, 0.5]), and network size (hses = [(200, 300), (300, 500)]), which includes the number of hidden units (hs) and the embedding size (es). For each model configuration (LSTM, Bi-directional LSTM, Bi-directional LSTM with Dropout), I selected the best parameters based on their performance in slot and intent classification.

There was a noticeable progressive improvement in both slot and intent classification as additional features were incorporated into the LSTM. Moreover, the best results were consistently achieved with the same parameter configuration: lr = 0.01, dp = 0.5, es = 300, and hs = 200.

Model	Parameters	Intent	Slot
LSTM	lr = 0.01, dp = 0.5, es = 300, hs = 200	0.92	0.917
LSTM+Bi	lr = 0.01, dp = 0.5, es = 300, hs = 200	0.935	0.946
LSTM+Bi+DROP	lr = 0.01, dp = 0.5, es = 300, hs = 200	0.953	0.941

Table 1: *Best results with 50 epochs*

3.2. Second phase

During the second phase, I mainly focused on optimising the learning rate (lr), dropout probability (dp) and the number of epochs, I selected the best parameters based on performance in slot and intent classification.

he BERT fine-tuning approach demonstrated the most substantial performance improvements, achieving the highest slot F1 scores and intent accuracy. This underscores BERT's superior ability to capture contextual information compared to traditional LSTM models, which is critical for the task. BERT's attention mechanism, enabling the model to focus on different parts of the input sequence, appears pivotal in delivering superior performance.

In addition to its performance advantages, the efficiency of BERT in training is noteworthy. Training LSTM models from scratch, particularly with added complexity like bidirectionality and dropout layers, is time-intensive.

In contrast, fine-tuning a pre-trained BERT model is notably faster. This efficiency accelerates training while yielding

improved performance metrics within a shorter training duration. These advantages highlight the practicality and effectiveness of utilizing pre-trained models such as BERT for natural language understanding tasks.

dp\epochs	10	30	50
0.1	intent=0.9395, slot=0.9628	intent=0.9361, slot=0.9742	intent=0.9406, slot=0.9745
0.5	intent=0.9395, slot=0.9735	intent=0.9361, slot=0.9753	intent=0.9765, slot=0.9720

Table 2: *Results with lr = 5e-4*

3.2.1. Others improvements

The performance of my model, when implemented using the parameters specified above but with a modified learning rate of 5e-5. The results obtained using my method are very similar to those obtained following the methodology proposed in the referenced paper. This adjustment to the learning rate was necessary due to the more complex tokenization process used in my method, which significantly hindered performance at higher rates.

dp\epochs	10	30	50
0.1	intent=0.9701, slot=0.9400	intent=0.9709, slot=0.9538	intent=0.9731, slot=0.9573
0.5	intent=0.9731, slot=0.9582	intent=0.9765, slot= 0.9535	intent=0.9765, slot=0.9601

Table 3: *Results with lr = 5e-5*

4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.