NLU course projects: lab 5

Simone Roman (mat. 247181)

University of Trento

simone.roman@studenti.unitn.it

1. Introduction

This assignment focuses on using a language model for intent classification and slot filling with the ATIS dataset. The first part of the assignment aims to enhance performance by applying two incremental changes: adding bidirectionality and dropout layers. In the second part, the objective is to adapt the code to fine-tune a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model. This requires modifying the dataset tokenization to align with the BERT tokenizer and adjusting the model to effectively perform both slot filling and intent classification.

2. Implementation details

Part one of the assignment required two major changes to the original model. First, I enabled bidirectionality in the LSTM, and also doubled the hidden size. Second, I added dropout layers after the embedding and LSTM layers. I fine-tuned the hyperparameters for optimal performance for each modification.

The second part of the assignmenet instead included the implementation of BERT with the aim of carrying out slot filling and intent classification, I took inspiration from the paper [1].

To achieve this, I first implemented BERT's pre-trained model. From BERT, I use two main outputs: the sequence output, which provides contextual representations for each token for slot filling, and the pooled output, which represents the [CLS] token for intent classification. These outputs are then passed to the respective linear layers: sequence output to the slot filling layer and pooled output to the intent classification layer, enabling the model to make both predictions effectively.

The second modification involves ID mapping, as BERT requires "[CLS]" at the start and "[SEP]" at the end, and it uses a Byte Pair Encoding (BPE) tokenizer that splits words into subwords, causing slot length discrepancies. To address these issues, I added "[CLS]" and "[SEP]" to the utterances, using "pad" for the corresponding slots. Then, I tokenized each word with BERT's tokenizer to ensure slot alignment. If a word splits into multiple tokens, the first token receives the original slot ID, while subsequent tokens receive "pad" IDs. This keeps slot and token sequences correctly aligned.

The final modification needed to adapt the original code for the BERT implementation was in the evaluation phase for slot filling. Specifically, when retrieving the outputs from the model and the ground truth, I had to remove the special "pad" tokens "[CLS]" and "[SEP]" and their corresponding positions from the model's output. These special tokens were initially added to meet BERT's requirements.

3. Results

For both parts of the assignment, the precision of intent classification was measured using accuracy, defined as the proportion of correctly classified intents out of the total. For slot filling, the F1-score was used, calculated considering only the slots with Inside-Outside-Beginning (IOB) tags. The results obtained in the assignment are shown in Table 1.

Additionally, there are two figures that show the graph of the loss during training compared to that of the evaluation. Figure 1 shows the results for the first part of the assignment, and Figure 2 shows the results for the second part.

Model name	Desctiption	Accuracy (%) Intent Classification	F1 Score (%) Slot Filling
model_11	Vanilla	93.72	91.93
model_12	Bidirectionality	94.85	94.05
model_13	Dropout	95.41	94.14
model_21	BERT	96.98	95.48

Table 1: *Results for all parts of the assignment. The first row shows the results with only hyperparameters modifications. The second row shows the results after adding bidirectionality. The third row shows the results after adding two dropout layers, one after the embedding layer and one after the linear layer. The fourth row shows the results for the second part of the assignment with the implementation of BERT. The first three results are cumulative, with each row building upon the previous optimizations.*

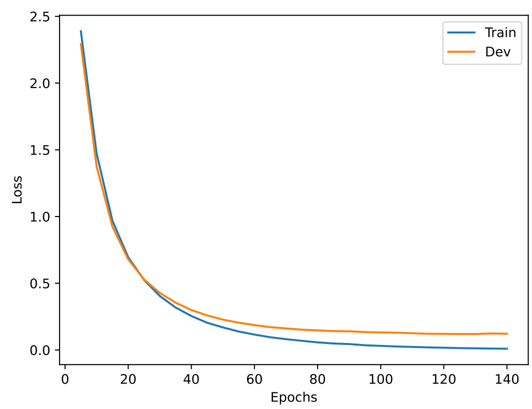

Figure 1: *Loss Training vs. Loss Validation for part 1 with the two optimizations (Bidirectionality, Dropout).*

4. References

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

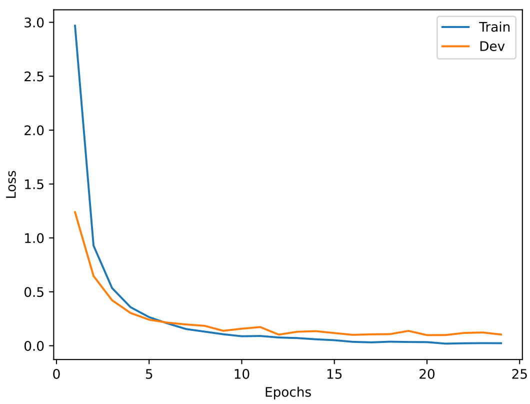

Figure 2: *Loss Training vs. Loss Validation for part 2 with the implementation of BERT.*