NLU project exercise lab: 10

Carlo Marotta (231590)

University of Trento

carlo.marotta@studenti.unitn.it

1. Introduction

This report explores how to make sequence labeling and text classification tasks more effective through advanced architectural improvements and multitask learning strategies. In the context of sequence labeling, we modified the *Model IAS* architecture by adding bidirectionality to better capture contextual cues and included a dropout layer to improve generalization. When it comes to text classification, we used the fine-tuned BERT model initialized with the pre-trained parameters in a multitask learning setting, addressing both intent classification and slot filling simultaneously. These enhancements highlight the potential of architectural adaptations and multitask learning for improving sequence labeling and text classification tasks.

2. Implementation details

In our initial two models, we implemented a series of modifications to enhance their performance. Initially, we introduced **bidirectionality** into the *LSTM* model, followed by the incorporation of a **dropout** layer. Our primary focus was on critical factors such as `LEARNING_RATE`, and in the case of the second model, `DROPOUT`. Throughout this process, we aimed to maximize *Slot F1* scores and *Intent Accuracy*, recognizing higher values as indicators of superior performance.
For the Model IAS with bidirectionality, we kept batch sizes and layer sizes consistent while experimenting with the `LEARNING_RATE` hyperparameter for the AdamW optimizer. Remarkable results were achieved by setting the `LEARNING_RATE` within the range of 0.0001 to 0.005. In the case of the second model with the dropout layer, we followed the same configurations as the first model, since we used the same vocabulary for training, maintaining a constant `DROPOUT` setting. We fine-tuned this setting to attain the highest accuracy, ultimately settling on a value of 0.3.
In the second scenario, involving the utilization of a pre-trained **BERT** model within a multitasking framework, we encountered several challenges in achieving satisfactory outcomes. Here as well, given that we employed a model initialized with pre-trained parameters, we concentrated exclusively on the `LEARNING_RATE` and `DROPOUT` parameters. The path to success entailed multiple iterations, primarily focused on reducing the `LEARNING_RATE` until a crucial observation emerged: the BERT model demonstrated efficient performance exclusively when this parameter was set within the open range of $(0.000001, 0.0001)$. Beyond this range, the performance remained notably unstable, yielding an Slot F1 score of 0.0. Subsequently, our attention shifted towards optimizing the dropout value, ranging from 0.5 to 0.05, to improve generalization. Ultimately, we determined that an optimal `LEARNING_RATE` of 0.00005, combined with a `DROPOUT` value of 0.1, led to significant performance enhancements, resulting in the highest *Intent Accuracy* among all the models discussed in this paper.

3. Results

For evaluating the model, I employed the central `train` function, where I initialized the model, optimizer, and loss functions. This function consisted of two crucial components: the training loop and the evaluation loop. The `train_loop` was responsible for overseeing the model's training process. It performed tasks such as calculating the loss, updating the model's parameters using the AdamW optimizer, and iteratively enhancing the model's performance. Simultaneously, the `eval_loop` was responsible for computing both *Slot F1*, which measures overall precision in detecting specific parts, and *Intent Accuracy*, which assesses how accurately the model grasps the primary goals.
To determine the most optimal model, I implemented an early stopping mechanism, configured with a `PATIENCE` parameter set to 20. This choice was influenced by the relatively fast training speed of the models, allowing for a maximum of 400 epochs. The purpose of the early stopping mechanism was to effectively mitigate the risk of overfitting. The `PATIENCE` value decreased each time the calculated perplexity exceeded its previous value. When the patience level reached 0, the training iteration was promptly halted.

3.1. Part 1

The provided data demonstrates that, in both scenarios, we achieved a commendable level of *Slot F1* and *Intent Accuracy*. Moreover, performance improved from the first model to the second one, which incorporated dropout. Our primary objective was to attain *Slot F1* and *Intent Accuracy* values surpassing the 90% threshold, where higher values indicate superior model performance. For the first model, we initially achieved approximately 80% accuracy in the first test. However, through parameter adjustments outlined in the previous section, we successfully boosted *Intent Accuracy* to approximately 95%. In the case of the second model, which introduced dropout, we consistently observed higher performance compared to the first model with the same parameter settings, eventually reaching an impressive 96% for *Intent Accuracy*.

3.2. Part 2

The incorporation of a pre-trained **BERT** model yielded a significant boost in accuracy for this case study. During model training, we utilized the AdamW optimizer, consistent with our previous models. Our initial attempts with learning rates both lower or equal to 0.000001 and higher or equal to 0.0001 did not produce conclusive results. Consequently, we narrowed our focus to the `LEARNING_RATE` range around 0.00005 and achieved the best results with this setting. In our pursuit of the optimal dropout configuration, we experimented with five different dropout values. Our findings revealed that a dropout value of 0.1 consistently delivered the highest performance, resulting in an impressive *Intent Accuracy* exceeding 97%.

4. Showing results

Table 1: *Part 1 and Part 2*

MODEL	BATCH_SIZE	EMB/HID_SIZE	LR	DROPOUT	**Slot F1**	**Intent Accuracy**
LSTM + bidirectional	128/64/64	300/200	0.001	-	0.94775	0.95744
LSTM + bidirectional + dropout	128/64/64	300/200	0.001	0.3	0.94749	0.96192
BERT	128/64/64	-	0.00005	0.1	0.92356	0.97312

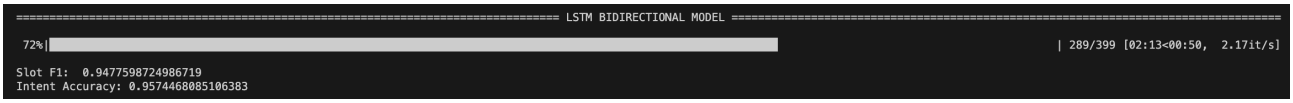

Figure 1: *LSTM bidirectional model*

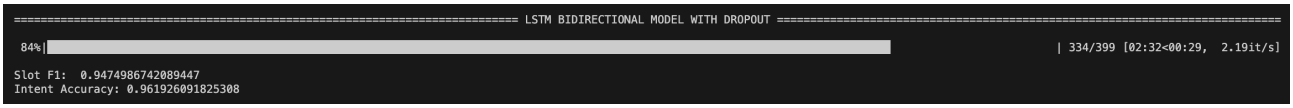

Figure 2: *LSTM bidirectional model with dropout*

Figure 3: *Fine-tuned BERT model with dropout*