NLU course project Part 2 - Natural Language Understanding

Alessia Ianes (256181)

University of Trento

alessia.ianes@studenti.unitn.it

1. Introduction

In this part of the project, the goal was to **enhance the performance of an existing intent classification and slot filling model** applied to the ATIS dataset. The first task involved modifying the baseline architecture by introducing **bidirectionality** and a **dropout layer** to enhance the model's generalization and accuracy. For the second task, the focus shifted to **fine-tuning a pre-trained BERT model** in a multi-task learning setup to handle both intent classification and slot filling simultaneously. This approach aimed to improve performance by leveraging the pre-trained knowledge of BERT, while addressing the challenges associated with sub-tokenization in fine-tuning. The performance of the models was evaluated using **intent classification accuracy and slot filling F1 score**, with results showing the impact of the applied modifications. *Parts of the code for this implementation were generated with the help of large language models (LLMs).*

2. Implementation details

The project aimed to improve intent classification and slot filling on the ATIS dataset using two models. In the first part, the baseline LSTM architecture was modified by introducing bidirectionality and a dropout layer to improve generalization. Then, it was trained using cross-entropy loss for both tasks and Adam as optimizer. In the second part, a pre-trained BERT model was fine-tuned for the same tasks in a multi-task learning setup. BERT's architecture was leveraged for token-level slot tagging, and its pooled output was used for intent classification. Sub-tokenization issues were addressed by aligning slot labels with BERT's tokenized outputs, with the **AutoTokenizer** from the Hugging Face library [1] used to tokenize the input sequences. Both LSTM and BERT models were evaluated using intent classification accuracy and slot filling F1 score, showing the advantage of fine-tuning BERT over the LSTM model.

3. Results

The initial LSTM-based model was enhanced by adding bidirectional layers [2], followed by the introduction of a dropout layer [3] after the embedding layer, the LSTM output, and the final hidden states. These modifications enabled the model to capture both past and future context for each token in the sequence and also to help prevent overfitting. Therefore, a phase of hyperparameter tuning was conducted by evaluating performance across distinct learning rate (from 1e-4 to 1e-3), batch size (32, 64, 128), hidden size (100, 200, 300) and dropout rate values (from 0.1 to 0.4). This optimization process resulted in a slight improvement, finding new best values for both accuracy (**from 0.94 to 0.96**) and F1 score (**from 0.93 to 0.95**), obtained using a learning rate of 1e-3, a batch size of 32, a dropout rate of 0.4 and a hidden size of 200. Figure 1 and Figure 2, which refer to the best results obtained with a particular configuration, show that the upgraded LSTM model (with bidirectionality and dropout) exhibits a more balanced reduction in both training and development losses compared to the baseline LSTM model. While the training loss continues to decrease steadily, the development loss also follows a similar trend, indicating that the model is better able to generalize to the validation data. In contrast, the baseline model shows a larger gap between training and development losses, suggesting overfitting. The addition of bidirectionality and dropout in the upgraded model helps mitigate this issue, improving generalization. The baseline model could also be more prone to overfitting, especially with the larger hidden and batch sizes tested.

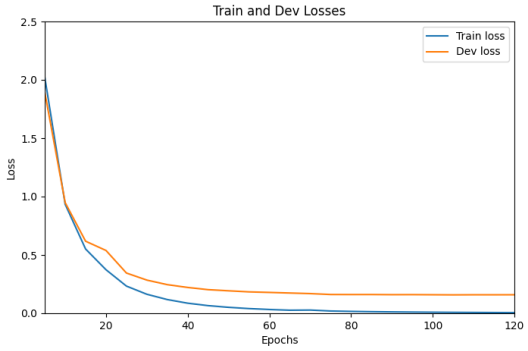

Figure 1: *Plot of loss function for training and validation set in LSTM baseline model*

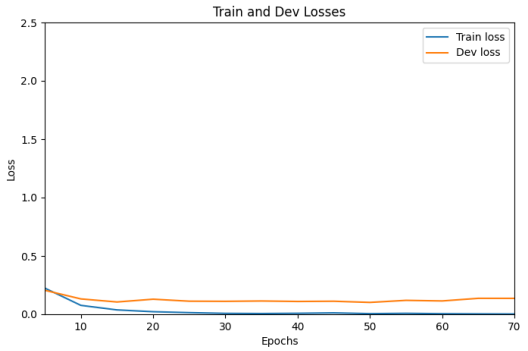

Figure 2: *Plot of loss function for training and validation set in LSTM model with bidirectionality and dropout*

Note that, in the end, also the other configurations yield very good results (greater than 0.93), as they differ only slightly from the best, as shown with another example in Figure 3.

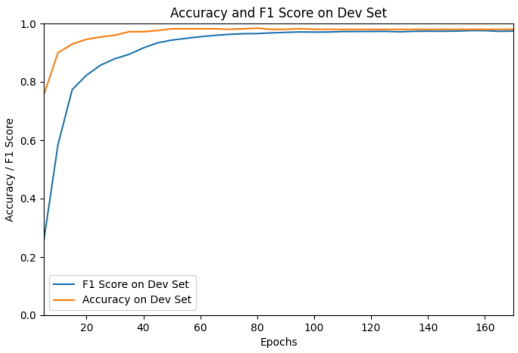

Figure 3: *Plot of validation slot filling F1 score and intent accuracy with a different configuration with respect to the one achieving the best results for this model*

However, despite these improvements, the model could not reach the performance levels of the BERT-based model, which benefits from a large amount of pre-trained knowledge. For this reason, the upgraded LSTM model was replaced by a BERT-based model, which - being pre-trained on a vast corpus of text, is capable of capturing nuanced language patterns that the LSTM model struggles to learn from scratch. The ability to handle sub-tokenization, through the alignment of slot labels with tokenized sequences, allowed for better slot tagging performance. To be more precise, in this part two types of BERT models were tested [4]: BERT-base-uncased and BERT-large-uncased. Despite BERT-large's capacity to model complex language relationships more effectively, the focus was put primarily on BERT-base since it requires lower resources, it is faster to train and it is generally less sensitive to hyperparameter choices, reducing the risk of training instability. Thus, another phase of hyperparameter tuning was conducted, where different learning rate, batch size and dropout rate values were evaluated. In particular, for BERT-large-uncased a single configuration was tested (learning rate of 1e-5, batch size of 32 and dropout rate of 0.1) obtaining very good results in F1 score (**0.94**) and accuracy (**0.97**). Regarding BERT-base-uncased, we used a setting similar to the one of the first part (same batch size and dropout rate values) with a difference in the learning rates used (from 1e-5 to 9e-5). This tuning yielded new best results in F1 score (**0.95**) and accuracy (**0.98**), achieved with a learning rate of 9e-5, a batch size of 128 and a dropout rate of 0.1. As, illustrated in Figure 4 and Figure 5, BERT-base model has a quicker convergence with a faster decrease in training loss, but it tends to overfit earlier, with the validation loss leveling off sooner. Instead, BERT-large model has a better generalization, with slower overfitting and improved validation loss in the initial epochs. However it has a higher computational cost and a greater tendency to overfit if not carefully regularized.

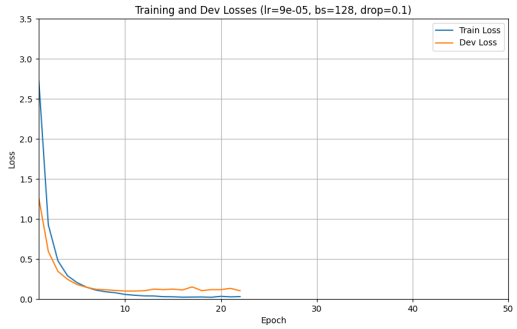

Figure 4: *Plot of loss function for training and validation set in BERT-base-uncased model*

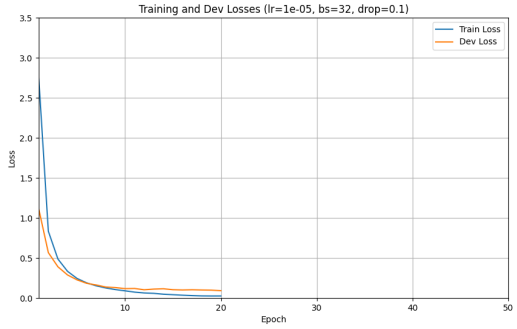

Figure 5: *Plot of loss function for training and validation set in BERT-base-uncased model*

In conclusion, the results demonstrate the effectiveness of fine-tuning pre-trained models like BERT for joint intent classification and slot filling tasks. While the LSTM model showed promising results with appropriate modifications, BERT's performance was superior, showcasing the advantage of using pre-trained models for such tasks.

	Model improvements	**Best F1 score**	**Best accuracy**
	Baseline LSTM	0.93	0.94
+	Bidirectionality	0.94	0.95
+	Dropout layers	0.95	0.96

Table 1: *Summary of the highest slot filling F1 scores and intent accuracies obtained for different configurations of the LSTM model during hyperparameter optimization in part 2A.*

Model improvements	**Best F1 score**	**Best accuracy**
BERT-large-uncased	0.94	0.97
BERT-base-uncased	0.95	0.98

Table 2: *Summary of the highest slot filling F1 scores and intent accuracies obtained for different BERT models during hyperparameter optimization in part 2B.*

4. References

[1] 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. [Online]. Available: https://arxiv.org/abs/1910.03771

[2] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805