NLU course projects: NLU

Rossana Cervesato (mat. 231499)

University of Trento

rossana.cervesato@studenti.unitn.it

1. Introduction

This is the final report for the Natural Language Understanding course, lab 5 (A.Y.2023/2024). The goal of the project is to fine-tune a pre-trained BERT model using a multi-task learning setting on intent classification and slot filling.
This project includes:

- the analysis of the ATIS dataset [1] (Table 1) and creation of the development set (1);
- the implementation and the training of the baseline and the proposed model;
- the proposed model's fine-tuning and the application of different architecture features;
- the evaluation of the models' behaviour.

The best model achieved an accuracy on intent classification of 97.53 and a F1 score in slot filling of 95.63 on the test set. The code is available at https://github.com/rossana24/NLU2024

2. Implementation details

2.1. Architecture

The ModelIASBaseline is a 2-layer bidirectional LSTM [2] with an embedding size of 300, a hidden size of 200. The baseline model also includes dropout regularization to embeddings and linear layer outputs.
The proposed model architecture uses a pre-trained BERT-base model [3, 4] as its core component for encoding input text sequences. The model also includes two linear classifiers with a linear layer and dropout: an IntentClassifier and a SlotClassifier. Additionally, a learning rate scheduler was employed to further refine the training process.
To prevent overfitting and enhance generalization, early stopping and weight decay were implemented in both baseline and proposed models.

2.1.1. Additional architecture features

The slot classification task often benefits from capturing both local patterns and sequential dependencies. Slots typically require detailed sequence information for accurate tagging, for this reason the proposed model architecture also investigates the possibility of adding a 1D-convolutional layer after the encoder network to capture local dependencies and patterns [5].
As the intent classification may depend on the slots in the sentence, another approach explored was the aggregation of slot-filling predictions from the slot classifier with the encoder's hidden representations to predict the overall intent of the utterance [5]. This is achieved through max pooling.

2.2. Dataloading and preprocessing

The use of a pre-trained BERT-base model within the proposed framework requires specialized data loading and preprocessing steps compared to the baseline. The workflow includes:

- Vocabulary specification for fair comparison with the baseline;
- Tokenization of the text and addressing the sub-tokenization problem;
- Utilization of the collate function to dynamically pad sequences, allowing for varying sequence lengths within each batch;
- Creation of attention masks to distinguish between real tokens and padding tokens.

A new vocabulary was created by considering only the tokens present in the training set, with any OOV tokens encountered in the dev and test sets being assigned an`<UNK>` token. Since BERT is already trained on a vast corpus, restricting the vocabulary to the training set ensures consistency in vocabulary usage between the baseline and the proposed model.
When sentences are tokenized for BERT, words may be split into sub-words, potentially disrupting the alignment between the original tokens and their corresponding slot labels. To address this, during dataset preprocessing, the slot logits are padded according to the size of the sub-tokens to ensure consistency between the lengths of the tokenized sentences and the slots.

2.3. Metrics and Evaluation

The models' performance was evaluated using two metrics: accuracy (2) for intent detection and F1-score (3) for slot filling.
After predicting slot labels, two challenges arise in evaluating the model: reversing tokenization to return to the original tokens in the dataset and removing additional padding not corresponding to the [SEP] and [CLS] tokens. While the latter is straightforward, detokenization requires a contraction map to handle English contractions, recombining tokens into their original forms (e.g., *I ' m* becomes *I 'm*).

3. Results and Discussion

Several experiments were conducted on the baseline to achieve the best configuration (Table. 2) (4).
The results obtained by the proposed model (Table. 3, Fig. 1, 2) demonstrates that the use of pre-trained bert model can result in competitive performance across multiple tasks.
The results also highlight the inherent trade-offs in multitask models. While the BERT model provides the best balance between accuracy and F1 score across both tasks, architectural modifications aimed at improving one task impacted the other. For instance, the aggregated approach, despite achieving the highest intent classification accuracy, resulted in a lower F1 score for slot filling. Similarly, the localized features detected by the convolutional layer might have introduced some noise into the predictions. Moreover, the small dataset size could have contributed to some variability in the results.

4. References

[1] Microsoft, "Atis dataset," https://github.com/Microsoft/CNTK/tree/master/Examples/LanguageUnderstanding/ATIS/Data, 2016, accessed: 2024-06-30.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," 2019. [Online]. Available: https://arxiv.org/abs/1902.10909

[4] monologg, "Jointbert," https://github.com/monologg/JointBERT, 2021, accessed: 2024-07-20.

[5] X. Qiu, Z. Zhou, Z. Chen, and X. Wang, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5467–5471.

Table 1: *ATIS dataset statistics.*

	Train set	**Test set**
# utterances	4978	893
# intents	22	20
# slots	123	101
Total intents	26	
Total slots	129	

1: *Details of the development set.*
The dev set is obtained by splitting the training data at 10%, ensuring that the dev set reflects the same intents and slots classes distribution as the training set. After the split, the final number of utterances is 4480 in the training set and 498 in the dev set.

2: *Accuracy equation:*

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

3: *F1 equation: where P is precision and R is recall*

$$F1 = 2 * \frac{P * R}{P + R}$$

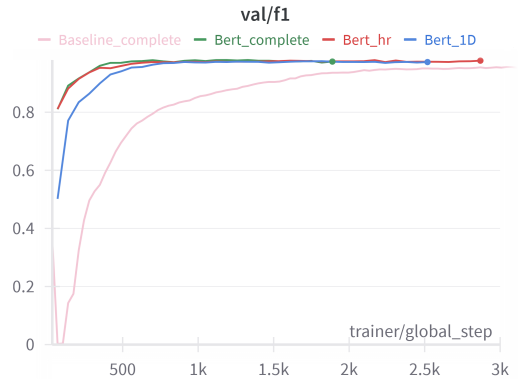
Table 2: *Baseline results. Where BD = bidirectional, Dp = additional Dropout, L{} = Number of layers.*

	Intent accuracy	**Slot F1**
Baseline_BD_L1	94.69 ± 0.41	93.24 ± 0.23
Baseline_BD_Dp_L1	95.14 ± 0.11	93.16 ± 0.30
Baseline_BD_Dp_L2	95.46 ± 0.34	92.93 ± 0.27

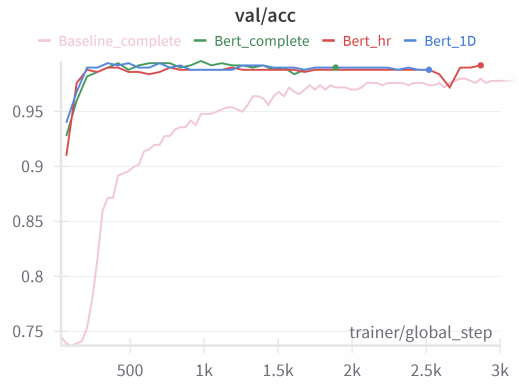
4: *Details of the experimental settings.*
Due to the small dataset size, the baseline models were trained five times, with the mean and standard deviation computed. In contrast, the proposed BERT model was trained only once due to the longer training times.

Table 3: *Proposed model results. Where 1D = additional convolutional layer, hr = improved hidden representation for intent prediction.*

	Intent accuracy	**Slot F1**
Bert	97.53	95.63
Bert_1D	97.64	95.11
Bert_hr	97.87	95.31



(a) *Validation F1 score for slot filling.*



(b) *Validation accuracy for intent classification.*

Figure 1: *Validation results.*

Figure 2: *Training results.*