# NLU course project - Lab 5

*Jonathan Fin (256178)*

University of Trento

jonathan.fin@studenti.unitn.it

## 1. Introduction

The objective of this project is to build a model to achieve the highest possible accuracy on **intent classification** and highest *F1 score* on **slot filling**. After being provided with a baseline LSTM, bidirectionality and a dropout layer are added to try to get higher scores (on accuracy and F1 score). For the second part of the project, the base LSTM is discarded and a fine-tuned version of a BERT model is trained for joint intent classification and slot filling.

## 2. Implementation details

The LSTM bidirectionality was added via a flag in the creation of the LSTM. The doubling in size was handled by multiplying by 2 the size of the hidden layer for the last linear layers and concatenating the last hidden states (both directions) before passing them to the last layers. The dropout was added after the embedding layer so that all of the network can benefit from it. Some tests were also run to confirm the correctness of this statement.

The second part of this project focused on fine-tuning a BERT model for intent classification and slot filling, as described in the paper *BERT for Joint Intent Classification and Slot Filling* [1]. A *JointBERT* model was created, which includes he base BERT model and two more linear layers for slot filling and intent classification. The main issue is the **subtokenization issue**, which happens when the classifier divides a word into subtokens. It is due to BERT's byte pair encoding. This is handled by forcing the subtokens (which start with `"##"`) to be transformed into `"[PAD]"`, which will be later ignored by the BERT classifier. An example can be found in Table 3.

Some other functions needed to be modified to adapt to the new model, such as the `eval_loop`, which skips the `[PAD]` token and uses the `BertTokenizer` instead of the `Lang` class to convert the ids to tokens. The model was inspired by *monologg*'s *JointBERT* [2], simplifying it to fit the task of this project.

## 3. Results

Each variation of the base LSTM in the first part of the project was evaluated on the **F1 score** for slot filling and **accuracy** for intent classification on the ATIS dataset. Table 1 highlights the best scores for each model.

| Model | Slot | Intent |
|---|---|---|
| Base LSTM | 92.0% | 92.9% |
| Bidirectional LSTM | 93.9% | 94.4% |
| B-LSTM with dropout | 94.7% | 95.3% |

Table 1: *LSTM models comparison*

Different numbers of layers were also tested, and as the quantity increased, the intent accuracy stayed the same and the slot F1 score decreased, due to overfitting. For example, with 5 layers in the LSTM, the F1 score was $91.9\%$; with 10 layers it was $88.4\%$.

The JointBERT model was tested on the same dataset. Firstly with no dropout and then with different dropout values. The dropout was added before the last linear layers. Table 2 shows the different results.

| Dropout probability | Slot | Intent |
|---|---|---|
| 0 | 90.4% | 95.6% |
| 0.1 | 92.8% | 96.5% |
| 0.3 | 92.7% | 96.2% |
| 0.5 | 92.7% | 95.7% |

Table 2: *JointBERT models comparison*

In conclusion, both LSTM and a fine-tuned version of BERT are good for slot filling and intent classification, with **Bert** (with dropout $= 0.1$) being the best among the two for intent classification and the **bidirectional LSTM** the best for slot filling.

## 4. References

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

[2] monologg, "Jointbert: Pytorch implementation of "bert for joint intent classification and slot filling"," https://github.com/monologg/JointBERT.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Original Tokens** | She | works | at | BioGenTech | Corporation | | | | |
| **Labels** | O | O | O | B-ORG | I-ORG | | | | |
| **BERT Subtokens** | [CLS] | she | works | at | bio | ##gen | ##tech | corporation | [SEP] |
| **Slots** | [PAD] | O | O | O | B-ORG | [PAD] | [PAD] | I-ORG | [PAD] |

Table 3: *Example of subtokenization and label alignment for BERT*