NLU course projects: Lab. 5 (NLU)

Andrea Richichi — University of Trento
andrea.richichi@studenti.unitn.it - 257850

1. Introduction

This report presents my work on two fundamental tasks in Natural Language Understanding (NLU): **intent classification** and **slot filling**. The goal of the first is to infer the user's overall intention from an utterance, while the second focuses on identifying and labeling key semantic components within it. Since these tasks are often closely intertwined, modeling them jointly enables the system to capture useful dependencies between global and token-level signals.

I explored two model families that reflect distinct approaches to sequence modeling:

- A joint model built around an LSTM encoder.
- A transformer-based model leveraging a pre-trained BERT backbone.

All experiments were conducted on the ATIS dataset, a well-established benchmark for spoken language understanding. To ensure reliable and stable evaluation, I averaged results over multiple runs and accounted for variance due to random initialization and data shuffling.

2. Implementation Details

In both models, training is guided by a joint objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{intent}} + \mathcal{L}_{\text{slot}}$$

Each term corresponds to a cross-entropy loss applied independently to intent labels and slot labels. I also used early stopping based on validation performance to prevent overfitting and encourage better generalization.

2.1 LSTM-based Joint Model

The first architecture uses an embedding layer followed by a bidirectional LSTM encoder. The final hidden state is used to predict the intent, while the full sequence of token-level hidden states is used for slot classification.

I experimented with two extensions to improve performance:

- **Bidirectional LSTM**, to capture context from both directions.
- **Dropout**, applied to embeddings and LSTM outputs ($p = 0.3$) to reduce overfitting.

The model is trained using the Adam optimizer with a learning rate of 0.001, and training stops early if no improvement is observed on the validation set for 5 consecutive epochs.

2.2 BERT-based Joint Model

The second approach fine-tunes the entire `bert-base-uncased` model. For intent classification, I use the final hidden representation of the `[CLS]` token. For slot filling, predictions are based on the token-level output of BERT.

Because BERT uses subword tokenization, I assign slot labels only to the **first sub-token** of each word. To handle this alignment, I rely on the `word_ids()` method from the Hugging Face tokenizer, and set the loss to ignore subtokens that do not correspond to word heads by assigning them an index of `-100`.

The model is trained end-to-end using the AdamW optimizer with the following hyperparameters:

- Learning rate: 5×10^{-5}
- Batch size: 64
- Dropout: 0.1
- Patience: 3

3. Results

I evaluated both models using two standard metrics:

- **Intent Accuracy**, which measures how often the predicted intent matches the ground truth.
- **Slot F1-score**, computed using the CoNLL evaluation script, which accounts for both token-wise precision and recall.

Among the LSTM variants, the BiLSTM with dropout produced the best results, offering clear improvements in both tasks compared to the simpler baseline.

The BERT-based model, on the other hand, consistently achieved the highest overall scores. Its ability to capture long-range dependencies and contextual nuances proved particularly beneficial for the more fine-grained slot filling task.

4. Conclusion

This project explored joint models for intent classification and slot filling using the ATIS dataset. The LSTM-based architecture demonstrated solid performance with relatively simple components, while the BERT-based model outperformed it by a significant margin, especially on the slot labeling task. These results highlight the strengths of transformer-based encoders in capturing complex semantic patterns across sequences, making them well-suited for joint NLU tasks.

Architecture	Intent Accuracy	Slot F1
ModelAS (LSTM)	0.9355 ± 0.003	0.9237 ± 0.004
ModelAS - BiLSTM	0.9474 ± 0.005	0.9409 ± 0.004
ModelAS - BiLSTM + Dropout	**0.9536 ± 0.002**	**0.9456 ± 0.002**
JointModel (BERT)	**0.9736 ± 0.004**	**0.9538 ± 0.001**

Table 1: Mean ± std over 5 runs. Best result among LSTM variants in gray, best overall in yellow.

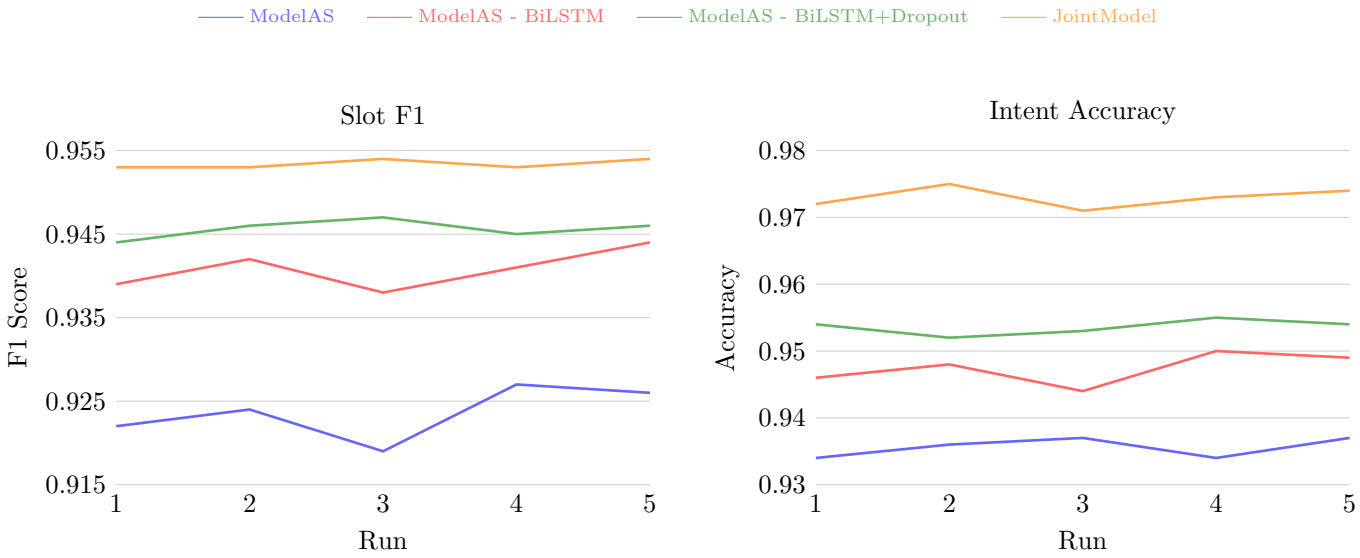

Figure 1: Performance comparison across 5 runs for intent classification and slot filling models.