NLU course project

Stefano Racca (mat. 256173)

University of Trento

stefano.racca@studenti.unitn.it

1. Introduction

In Part A, I improved a baseline unidirectional LSTM model for ATIS joint intent classification and slot filling by replacing the encoder with a bidirectional LSTM and adding a 0.2 dropout layer on embeddings and encoder outputs. I evaluated four configurations that varied learning rate (0.001 vs. 0.0001) and embedding/hidden-size (200/300 vs. 128/100), each trained with early stopping and averaged over five runs.

In Part B, I fine-tuned two pre-trained BERT variants (`bert-base-uncased` and `bert-large-uncased`) in a multi-task setting. A shared BERT encoder feeds separate heads for intent (pooled [CLS]) and slot (token-level). Word-level slot tags are aligned to subtokens using the tokenizer, assigning the true label to the first subtoken and an ignore index (–100) to subsequent subtokens. Both parts use the standard ATIS splits to measure slot F1 and intent accuracy.

2. Implementation details

Part A. Starting from a baseline Model IAS (unidirectional LSTM over pre-trained embeddings), I implemented:

- *Bidirectional LSTM*: I replaced the LSTM with a bidirectional variant. For each time step, I concatenated forward and backward hidden states, yielding richer contextual representations.
- *Dropout Layers*: I applied dropout (probability 0.2) immediately after the embedding layer and after concatenating bidirectional hidden states to reduce overfitting on ATIS's small training set.

I ran four configurations:

1. *Bidirectional-0.001*: hidden size 300, no dropout, learning rate 0.001.
2. *Bidirectional-0.0001*: same architecture with learning rate 0.0001.
3. *Bidirectional + Dropout-1* : embedding 200, hidden 300, dropout 0.2.
4. *Bidirectional + Dropout-2*: embedding 128, hidden 100, dropout 0.2.

I trained each for up to 50 epochs using Adam (learning rates as above), batch size 32, and early stopping (patience = 2, but update every 5 epochs, so 10 in total) based on dev-set slot F1. Slot tags used IOB format and were aligned to token outputs.

Part B (BERT Fine-tuning). For each BERT variant, I loaded the pretrained model and tokenizer from Hugging Face. Key steps:

- *Subtoken Alignment*: Using `is_split_into_words=True`, the tokenizer returns a `word_ids()` mapping. I assigned each first subtoken its word's slot label (from `Lang.slot2id`) and marked subsequent subtokens with -100 so they are ignored by the slot loss.
- *Model Architecture*: The shared BERT encoder produces:
  - **Sequence Output** ($[B, T, H]$) for token-level slot prediction (linear layer $\rightarrow$ # slots).
  - **Pooled Output** ($[B, H]$) for intent classification (linear layer $\rightarrow$ # intents).

  I applied dropout (0.1) after BERT outputs.
- *Training*: I fine-tuned for up to 100 epochs using AdamW (learning rate 2×10^{-5}, weight decay 0.01), batch size 32, and early stopping on dev-set slot F1 (patience = 3, here updated every epoch). Loss was the sum of cross-entropy intent loss and slot loss (with ignore index –100). Gradients were clipped at 1.0. I saved the checkpoint with highest dev slot F1 and evaluated on the ATIS test set using the CoNLL script for slot F1 and intent accuracy.

This design and training procedure were inspired by the multi-task learning formulation for joint intent and slot modeling introduced by Qian et al. [1].

3. Results

In Part A, I replaced the unidirectional LSTM with a bidirectional one and I obtained a slot F1 of approximately 0.94 and 0.945 for intent accuracy. Lowering the learning rate to 0.0001 slowed convergence, yielding slightly lower F1 (0.934) and accuracy (0.941). Adding dropout (0.2) on both embeddings and encoder outputs raised slot F1 to about 0.945 and intent accuracy to 0.951. The compact configuration (emb = 128, hid = 100) matched the larger model's slot F1 (0.94539 vs. 0.94543) and attained the highest intent accuracy (0.953), demonstrating that a smaller architecture with dropout can achieve comparable performance while reducing parameters.

In Part B, fine-tuned `bert-base` outperformed all LSTM variants, achieving slot F1 = 0.959 and intent accuracy = 0.976. Surprisingly, `bert-large` did not surpass `bert-base`, with slightly lower scores (slot F1 = 0.957, intent accuracy = 0.971). This underperformance may be due to overfitting, given the limited size of the ATIS training set. However, both BERT variants clearly outperformed LSTM configurations by 1–1.5% in slot F1 and 2–3% in intent accuracy, confirming the advantage of pre-trained contextual embeddings. Notably, `bert-large` reached its best performance in only 15 epochs, compared to 29 for `bert-base`, suggesting faster convergence despite its higher capacity.

I report token-level slot F1 (CoNLL script) and intent accuracy on the ATIS test set. Table 1 gives Part A's averaged results (five runs); Table 2 shows Part B's test-set metrics (best checkpoint).

Table 1: *Part 2.A: Bidirectional LSTM Configurations on ATIS (mean over five runs)*

Experiment	**Slot F1**	**Intent Accuracy**
Bidirectional-0.001	0.94015	0.94513
Bidirectional-0.0001	0.93404	0.94199
Bidirectional + Dropout-1	0.94543	0.95162
Bidirectional + Dropout-2	0.94539	0.95364

Table 2: *Part 2.B: Fine-Tuned BERT on ATIS*

Model	**Epochs Trained**	**Slot F_1**	**Intent Acc.**
`bert-base`	29	0.95896	0.97648
`bert-large`	15	0.95669	0.97088

4. References

[1] W. W. Qian Chen, Zhu Zhuo, "Bert for joint intent classification and slot filling," 2019.