NLU course project - Lab5

Mauro Antonio de Palma (256175)

University of Trento

mauroantonio.depalma@studenti.unitn.it

1. Introduction

This report shows the results of the experiments in the context of **Slot filling** and **Intent classification**. These two concepts are bound together in a NLU scenario where *Intent classification* allows the software to understand the main user intention while *Slot filling* extracts the parameters related to the intention by mapping utterance tokens to domain-specific labels.

- *PartA* enhances an existing LSTM architecture adding bidirectionality and a dropout layer after the initial embedding layer and before the output layer.
- *Part B* fine-tunes the BERT[1] model taking advantage of its pre-trained weights.

2. Implementation details

Both experiments use the ATIS (Airline Travel Information Systems) dataset. To avoid being too dependant on initial random weights the calculated metrics will be averaged across 5 runs of 100 epochs using AdamW optimizer (learning rate = 0.0001). Models are evaluated using **F1-Score** for slot filling and **Accuracy** for intent classification. The training objective is to minimize their **CrossEntropy** loss combined using *Uncertainty weight*[2] using the following formula:

$$\mathcal{L} = \frac{1}{\sigma_i^2}\mathcal{L}_i + \frac{1}{\sigma_s^2}\mathcal{L}_s + \log(\sigma_i^2) + \log(\sigma_s^2) \qquad (1)$$

Where $\log(\sigma_i^2)$ and $\log(\sigma_s^2)$ are two learned parameters. $s = \log(\sigma^2)$ is used to ensure numerical stability (avoids 0-division and negative values for the variance) since $\sigma^2 = e^s$. This joint loss will be used as training/validation loss.

2.1. Part A

The baseline LSTM has been enhanced by incrementally applying some new design approaches and regularization techniques:

- *Bidirectionality*: In natural language each word takes its meaning from other words in both directions. Letting the model process data in both forward and backward directions allows it to get the full context and knowledge using all the available data
- *Dropout*: Dropout randomly drop neurons with a probability *p = 0.3* only at training time. This is useful to add uncertainty during the training time and avoid being stuck in local optima. This layer will be applied after the embedding and before the output layers.

2.2. Part B

This part fine-tunes the pre-trained BERT model for joint intent classification and slot filling. Both BERT-base (768-dimensional hidden states) and BERT-large (1024-dimensional) are evaluated.

Intent classification: the model leverages *[CLS]* token representation from BERT's final transformer layer which encodes the entire utterance semantic meaning. A linear layer projects this 768-dimensional vector (1024 for BERT-large) to the 21 ATIS intent classes.

Slot Filling: To handle BERT's *WordPiece* sub-tokenization, the preprocessing assigns each word's slot label only to its *first* sub-token. Additional sub-tokens generated by WordPiece are padded with a special `pad` token. During training, `CrossEntropyLoss` ignores these padded positions. At evaluation time, only positions with non-pad ground truth labels are considered for the F1-score calculation, ensuring alignment between predictions and gold labels.

3. Results

The metrics evaluated are the accuracy for the intent classification and the F1 score for slot filling using the test dataset (never used during the training phase) Table 1 and Table 2 show the results of all the experiments conducted in the first and second part. Image 1 and Image 2 show an interesting chart about how the weight to combine slots/intents losses are learnt during the epoch. Another important data (as you can see by 2) is that the BERT based model reaches stable metrics in less epochs.

- **Part A**: The enhancements (bidirectionality and dropout) provide an incremental improvement in F1-score and accuracy over the baseline.
- **Part B**: Fine-tuning BERT yields significant performance gains, with BERT-Large offering a slight edge over BERT-Base due to its increased capacity.

Architecture	Slots F1	Intents Acc.
LSTM	0.8983 ± 0.0061	0.9342 ± 0.0023
LSTM bidirectional	0.93 ± 0.0033	0.9445 ± 0.0017
LSTM bidirectional + dropout	0.926 ± 0.0029	0.951 ± 0.0013

Table 1: *LSTM model with bidirectionality and dropout*

Architecture	Slots F1	Intents Acc.
BERT-base	0.9540 ± 0.0025	0.9760 ± 0.0017
BERT-large	0.9571 ± 0.0022	0.9722 ± 0.0019

Table 2: *BERT fine-tuning results*

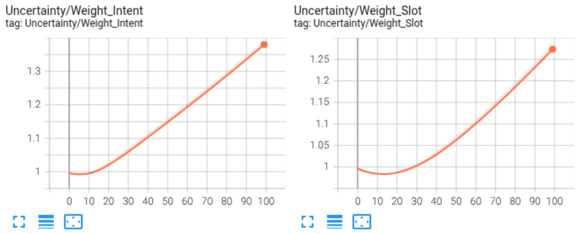

Figure 1: *Evolution of uncertainty weights during training.*

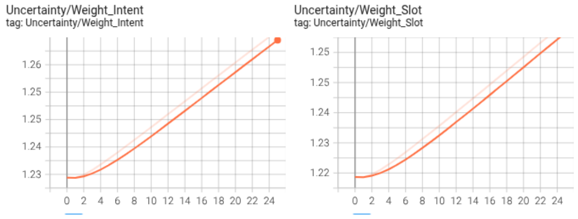

Figure 2: *Evolution of BERT uncertainty weights during training.*

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] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” *arXiv preprint arXiv:1705.07115*, 2017. [Online]. Available: https://arxiv.org/abs/1705.07115