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 by initial random weights the calculated metrics will be averaged across 5 runs of 100 epochs using AdamW optimizer (learning rate = 0.0001). The loss is calculate as follows:

- **F1-Score** for SlotFilling task
- **Accuracy** for Intent classification

These two metrics will be combine by using *Uncertainty weight loss*[2] using the following formula:

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

Where $log(\sigma_i^2)$ and $log(\sigma_s^2)$ are two learned parameters to find the best compromise between to two losses (for convenience I used log to avoid 0-division since $x = e^{-log(x)}$

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 to 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

In this second part we'll rely on the pre-trained model BERT. We will evaluate both the base and large model showing at the end the performances gains. The *intent classification* we'll take benefit of the [CLS] representation after the final transformer layer. An output linear layer will map this knowledge from a 768-dimensional space (1024 for BERT large) into the 21 ATIS intent classes. For the *slot filling*, instead, it's needed to deal with the sub-tokenization issue: BERT uses WordPiece tokenization, which can break words into smaller subword units, especially for rare or complex words, resulting in a problem of label alignment and ambiguity. To address this, the implemented solution maps slot labels only to the first sub-token of each word while maintaining consistency across sentences. In particular, the method uses the tokenizer from AutoTokenizer [2] that allow to map tokens back to their original words through the word ids() function [3]. This function provides a list in which each token is assigned an ID corresponding to its original word in the sentence. If a word is split into multiple tokens, all these tokens are assigned the same ID, helping us identify all parts of a word. To ensure that only the first sub-token of each word is mapped to a slot label, keeping the alignment straightforward the solution also check for spaces between words, so it's possible to confirm when a new word starts and align it correctly. By creating a mapping that links each word with just its first token, it ensures that slot labels correspond clearly to each word. This method allows maintaining consistency in slot labeling, even with sub-tokenized words, ultimately improving the model's ability to correctly assign slot labels across different inputs.

3. Results

Add tables and explain how you evaluated your model. Tables and images of plots or confusion matrices do not count in the page limit.

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*.