NLU course projects

Nicola Maestri (239920)

University of Trento

nicola.maestri@studenti.unitn.it

1. Introduction

In this assignment, we develop a *language model* (LM) for joint *intent classification* and *slot filling*. Initially, we implement an *LSTM*-based architecture, followed by separate classifiers for intent and slot predictions. Performance improves using a *bidirectional LSTM* and adding *dropout layers*. Integrating *BERT* as a foundation model and leveraging its contextual embeddings further improves test scores by over $> 2.5\%$ in both tasks. Finally, we evaluate the impact of fine-tuning all *BERT* layers, half of them and only the last one, observing a performance drop in *slot filling* for reduced fine-tuning.

2. Implementation details

In the first experiment, we use an *LSTM* followed by separate classifiers for intent and slot predictions. Additional enhancements include bidirectional LSTMs and dropout layers applied to embeddings and hidden states. Network hyperparameters are detailed in Table 1.

In the second experiment, *BERT* is adapted for intent and slot classification, as outlined in [1]. The *CLS* embedding is passed through a dropout layer and a linear classifier for intent classification. Similarly, the last hidden states are passed through a dropout layer and a linear classifier for slot filling. To address sub-tokenization, only the first sub-token of each word is used for slot classification. Slot sequences are padded and aligned with the first sub-tokens, ignoring *pad* tokens in the loss function. Figure 1 illustrates this alignment. We experiment with *BERT-base* and *BERT-large* models, exploring full finetuning, partial finetuning of the lower layers, and finetuning only the last layer.

In both experiments, training proceeds for a maximum of 200 epochs, with early stopping triggered after 5 consecutive validation stagnations. We use *AdamW* as the optimizer and clip gradients to ensure stability. A larger learning rate ($1e-4$) is applied to the classifiers, while a smaller rate ($5e-5$) is used to fine-tune *BERT*. Configuration details are provided in Table 2.

Intent Classification and *Slot Filling* tasks employ the *Cross Entropy* loss function, excluding <*pad*> tokens. The losses are summed and minimized jointly. Intent classification is evaluated using accuracy, while slot filling is measured using the *CoNLL* script to report the F1 score.

We use the *ATIS* dataset for training and evaluation. For experiments involving the *LSTM* model, a vocabulary is created to map words to IDs. For *BERT*-based models, we employ the *BERT-tokenizer*. Specifically, we consider *BERT-base* and *BERT-large* as backbones. Parameters for the data loaders are summarized in Table 3.

3. Results

In all experiments, we report the average score over 5 runs along with the relative standard deviation. Results from both experiments are summarized in Table 4 and illustrated in Figures 2 and 3.

Our baseline achieves 93.5% on Intent Classification and 91.1% on Slot Filling. Adding bidirectionality results in a $> 1.5\%$ improvement in both tasks, while introducing dropout has minimal impact on performance, though it reduces the standard deviation. Figures 4 and 5 show the progression of intent accuracy and slot F1 scores on the validation set.

Using *BERT* as the backbone significantly enhances performance, with the best settings achieving 97.5% accuracy on Intent Classification and 96.1% on Slot Filling. Notably, fine-tuning only the last layer surpasses the performance of the *LSTM*-based model in both tasks. Results are comparable between *BERT-base* and *BERT-large*, with the best performance obtained by fine-tuning the entire model. Figures 4 and 5 further depict the trends in intent accuracy and slot F1 scores on the validation set.

4. References

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

A. Tables and Figures

LSTM hyperparameters	
Embedding size	300
Hidden size	200
Slot Classifier	130
Intent Classifier	26
n layers	1
pad index	0
Output dropout	0.1
Embedding dropout	0.1

Table 1: *LSTM hyperparameters*

Training hyperparameters	
n epochs	200
n epochs for fine-tuning	50
patience	5
clip	5
training loss	*Cross Entropy*
Intent evealuation metric	*Accuracy*
Slot filling metric	*F1 score*
optimizer	AdamW
learning rate	1 e-4
learning rate bert parameters	5 e-5
weight decay	0.01

Table 2: *Training hyperparameters*

Training loader	
Length training dataset	4480
Batchsize	128
Number of batches	35
Validation loader	
Length validation dataset	498
Batchsize	64
Number of batches	8
Test loader	
Length test dataset	893
Batchsize	64
Number of batches	14

Table 3: *Loader hyperparameters*

Model	Intent		Slot filling	
	Accuracy	std	F1-score	std
LSTM	93.5	0.6	92.1	0.4
LSTM bidirectional	**94.8**	0.5	93.7	0.2
LSTM bidir + dropout	94.7	0.1	**93.9**	0.1
Bert_base fully tuned	**97.5**	0.3	95.7	0.1
Bert_base partially tuned	97.4	0.3	95.4	0.1
Bert_base 1 layer	97.2	0.1	94.0	0.9
Bert_large fully tuned	97.4	0.2	**96.1**	0.2
Bert_large partially tuned	97.4	0.2	95.7	0.1
Bert_large 1 layer	96.7	0.2	91.9	0.3

Table 4: *NLU performance on ATIS dataset. The metrics are intent classification accuracy and slot filling F1 (%). Results are averaged over 5 runs with relative standard deviation (std).*

Original Tokens	First sub-token	Slot Labels	Slot IDs
[CLS]	- - -	<pad>	0
show	show	O	43
me	me	O	43
the	the	O	43
cheap	cheap	B-cost_relative	35
##est	- - -	<pad>	0
flight	flight	O	43
from	from	O	43
pittsburgh	pittsburgh	B-fromloc.city_name	46
to	to	O	43
atlanta	atlanta	B-toloc.city_name	12
on	on	O	43
wednesday	wednesday	B-depart_date.day_name	74
which	which	O	43
leaves	leaves	O	43
before	before	B-depart_time.time_relative	117
noon	noon	B-depart_time.time	37
and	and	O	43
serves	serves	O	43
breakfast	breakfast	B-meal_description	76
[SEP]	- - -	<pad>	0

Figure 1: *Tokenized utterance alligned with slot labels.*

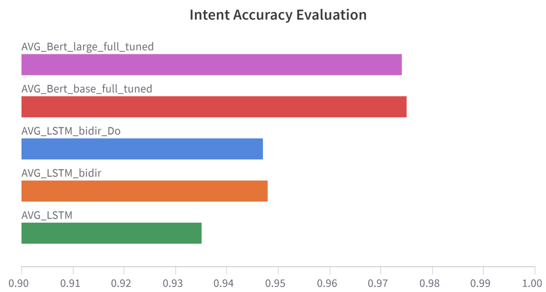

Figure 2: *Comprehensive comparison*

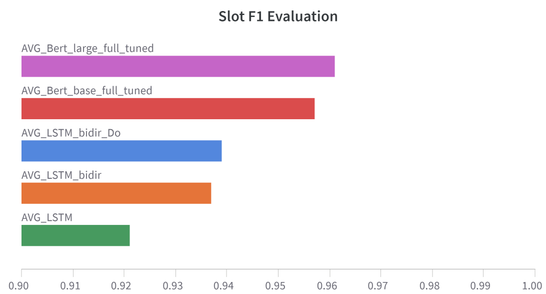

Figure 3: *Comprehensive comparison*

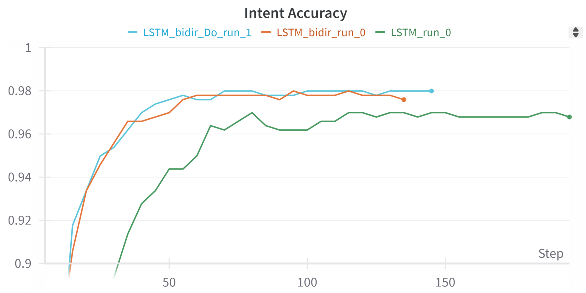

Figure 4: *Intent Accuracy on validation set in experiment 1*

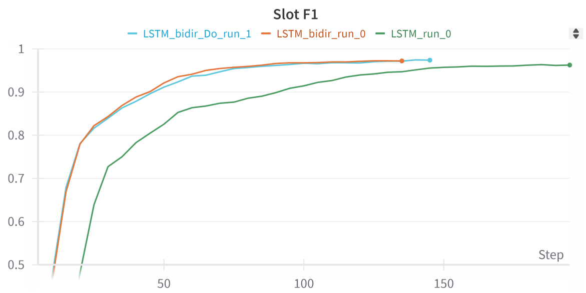

Figure 5: *Slot F1 on validation set in experiment 1*

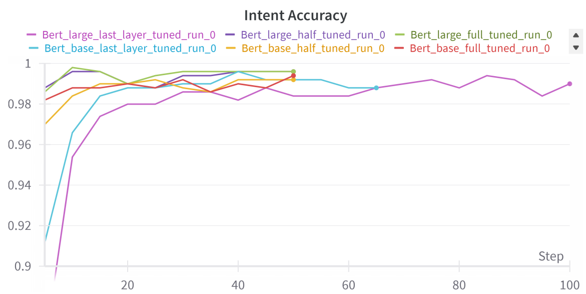

Figure 6: *Intent Accuracy on validation set in experiment 2*

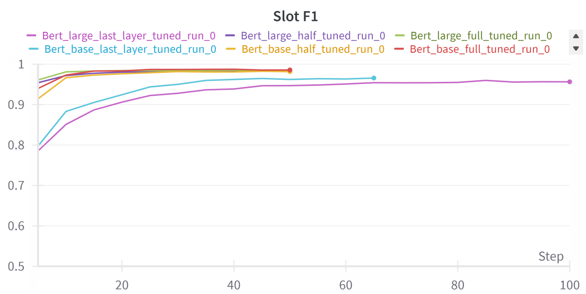

Figure 7: *Slot F1 on validation set in experiment 2*