NLU course projects - Assignment 2 - NLU

Ettore Saggiorato (247178)

University of Trento

ettore.saggiorato@studenti.unitn.it

1. Introduction

This assignment focuses on performing slot filling and intent prediction using the ATIS dataset. The task is divided into two parts:

Training a Long Short-Term Memory (LSTM) model and progressively enhancing it by incorporating bidirectionality and dropout layers to evaluate their impact on performance.

Fine-tuning a pre-trained BERT model [1] for the same task. The second part introduces additional complexity due to challenges associated with BERT's tokenizer.

Model performance is evaluated using two key metrics: accuracy for intent classification and F1 score for slot filling.

2. Implementation Details

2.1. Part 1 - LSTM Model

In the first part, we explore the impact of architectural enhancements—bidirectionality and dropout—on the performance of an LSTM model. These enhancements are applied sequentially to observe their contributions.

2.1.1. Architecture

Bidirectionality: Bidirectional LSTM is implemented using PyTorch's built-in LSTM module, which processes input sequences in both forward and backward directions. This configuration effectively doubles the hidden size, enabling the model to capture richer contextual information. Bidirectionality is particularly beneficial for understanding phrases with critical information occurring towards the end of a sentence, as it allows the model to integrate context from both directions.

2.1.2. Regularization

Dropout: Dropout is applied after the embedding layer to enhance the model's generalization capabilities. By randomly deactivating neurons during training, dropout forces the model to learn more robust features instead of relying on specific patterns. This makes the model more resilient to poorly structured or noisy inputs.

2.2. Part 2 - BERT Fine-tuning

The second part of the assignment involves fine-tuning a pre-trained BERT model [1] for intent prediction and slot filling. This process follows the guidelines outlined in BERT's original paper [1], particularly regarding hyperparameter settings such as learning rate, batch size, and the number of training epochs. Each experiment is conducted over four epochs and repeated five times to ensure consistent results through averaging.

The `bert-base-uncased` model and tokenizer are used. However, BERT's tokenizer poses challenges, as it does not natively support the tags required for intent classification and slot filling. To address this issue, three approaches were explored:

1. **Extending the Tokenizer Vocabulary:** This approach involved adding the required tags directly to the tokenizer's vocabulary. While this necessitated modifications to the model's input structure, such as altering the embedding layer or introducing a linear layer before it, the method proved ineffective. Additional training time might have yielded better results, but due to its complexity, this approach was abandoned.
2. **Tags Cloning:** The TokenID of the first token in a word is cloned for all the subtokens of that word. While conceptually straightforward, this method introduced significant distortions during loss computation and evaluation. Words split into multiple subtokens could incorrectly receive disproportionately positive or negative rewards, compromising model performance.
3. **Tag to Pad:** In this approach, only the first subtoken of a word was assigned the relevant tag ID, while all subsequent subtokens were treated as padding (ID = 0). This effectively eliminated the evaluation distortion seen in the cloning approach, ensuring more accurate training and assessment.

2.2.1. Training Regularization

To enhance training stability and mitigate overfitting, the following techniques were applied:

- **Gradient Clipping** used to prevent exploding gradients.
- **Learning Rate Scheduler.**

3. Results

3.1. LSTM Model

The results for the LSTM model are summarized in Table 1. The experiments demonstrate that bidirectionality significantly improves performance, enabling the model to capture richer contextual information and accelerating convergence. Adding dropout further enhances robustness, particularly in handling noisy or ambiguous input sequences.

3.2. BERT Fine-tuning

As suggested in the BERT paper's appendix [1], the model was finetuned both with three different lr (5e-5, 3e-5, 2e-5), two different batch sizes (16, 32), 4 epochs and the AdamW optimizer. The results, shown in Table 2, clearly demonstrate the positive impact of gradient clipping on training. However, the learning rate scheduler appears to negatively affect performance. Additionally, as said in Appendix A of BERT's paper, the model struggles with larger batch sizes, leading to less optimal convergence. More epochs could lead to performances, but more regularization, such as dropout is needed to avoid overfitting.

4. References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

Part	**Model Name**	**LR**	**Epochs**	**F1 Slot**	**Accuracy Intent**
1.0	baseline-1	0.0001	25	0.02	0.70
1.0	baseline-2	0.001	195	0.92	0.92
1.0	baseline-3	0.01	100	0.91	0.91
1.1	bid	0.001	95	0.93	0.94
1.2	drop01	0.001	95	0.90	0.92
1.2	drop02	0.001	95	0.89	0.93
1.3	**bidrop**	0.001	95	**0.94**	**0.95**

Table 1: *LSTM Experiment Results.* ***Bold*** *values represent the best peroformant model.*

Part	Model name	Epochs	Batch size	Gradient clip.	Train. sch.	F1 (%)	Accuracy (%)
2.0	simple_16_2e-5	4	16	F	F	0.60	0.90
2.0	simple_16_3e-5	4	16	F	F	0.71	0.88
2.0	**simple_16_5e-5**	4	16	F	F	**0.72**	**0.91**
2.0	simple_32_2e-5	4	32	F	F	0.27	0.82
2.0	simple_32_3e-5	4	32	F	F	0.39	0.87
2.0	simple_32_5e-5	4	32	F	F	0.36	0.89
2.1	grad_16_2e-5	4	16	T	F	0.70	0.90
2.1	grad_16_3e-5	4	16	T	F	0.74	**0.92**
2.1	***grad_16_5e-5***	4	16	T	F	**0.78**	**0.92**
2.1	grad_32_2e-5	4	32	T	F	0.38	0.81
2.1	grad_32_3e-5	4	32	T	F	0.56	0.89
2.1	grad_32_5e-5	4	32	T	F	0.67	0.91
2.2	sch_grad_16_2e-5	4	16	T	T	0.35	0.85
2.2	sch_grad_16_3e-5	4	16	T	T	0.51	0.89
2.2	**sch_grad_16_5e-5**	4	16	T	T	**0.66**	**0.90**
2.2	sch_grad_32_2e-5	4	32	T	T	0.26	0.73
2.2	sch_grad_32_3e-5	4	32	T	T	0.42	0.80
2.2	sch_grad_32_5e-5	4	32	T	T	0.35	0.89

Table 2: *Bert Experiment Results.* ***Bold*** *values represent the most performant models for each part, italics represent the best overall.*