NLU Course Project - Natural Language Understanding

Christian Dalvit (249988)

University of Trento

christian.dalvit@studenti.unitn.it

1. Introduction

This project investigates the joint learning of intent classification and slot filling for natural language understanding using Long Short-Term Memory (LSTM) units and pre-trained BERT models as backbones. The LSTM-based models are enhanced with bidirectional architecture and dropout layers, while the BERT model is fine-tuned to improve performance. The effectiveness of these techniques is evaluated on the ATIS dataset using the F_1 score as a metric. The code of this project is made available on Github.

2. Implementation details

All models in this project are implemented in PyTorch [1]. The LSTM [2] implementation from the `ModelIAS` class provided in Lab 5 is used as baseline. The baseline consists of an embedding layer, an LSTM unit, and a linear output layer, with the Adam optimizer employed for training. For the BERT [3] implementation, code snippets provided in Lab 5 are used as a reference.

First Part In the first part, the baseline model is extended by adding bidirectionality to the LSTM unit. PyTorch provides a flag in its LSTM implementation to enable bidirectionality. When using a bidirectional LSTM, the input size of the linear output layer for slot filling must be doubled. Additionally, dropout layers are added before the embedding layer and the linear layer for intent classification. As PyTorch includes built-in implementations for dropout layers, their integration is straightforward.

Second Part In the second part, pre-trained BERT models [3] are fine-tuned for slot filling and intent classification. The pre-trained models are accessed through the Huggingface library. Since Huggingface provides a convenient Python API, loading the different models was straightforward to implement. The project implementation supports loading weights for $\text{BERT}_{\text{LARGE}}$, $\text{BERT}_{\text{BASE}}$, $\text{BERT}_{\text{MEDIUM}}$, $\text{BERT}_{\text{SMALL}}$, $\text{BERT}_{\text{MINI}}$, and $\text{BERT}_{\text{TINY}}$ [3, 4]. All models process uncased input text. An additional dropout layer is implemented and can optionally be applied before the final linear layer. A primary challenge when using models from the BERT family is tokenization. BERT models employ WordPiece [5] tokenization, which can split a single input word into multiple subtokens. This creates a challenge for the slot filling task, as subtokens generates a corresponding hidden state. However, since slot filling requires only one label per word, it becomes necessary to either reduce the number of hidden states or adjust the labeling scheme to fit the subtokens. To address this, the approach proposed by Chen et al. [6] is adopted in this project: only the first subtoken is used for slot filling. This is implemented by filtering the tokenized input text to select the first subtoken of each word.

3. Results

All models were evaluated on the ATIS dataset, with the F_1 score used to measure slot filling performance and accuracy used for intent classification. The evaluations were conducted on the Marzola cluster at the University of Trento.

First Part In the first part, the baseline LSTM implementation was benchmarked against the bidirectional LSTM and the bidirectional LSTM with dropout. For all benchmarks in the first part, a hidden size of 200, an embedding size of 300, and a training batch size of 64 were used. Additionally, all LSTMs consisted of two layers, and training was performed over 200 epochs. No early stopping was applied for the first 5 epochs. After that, an early stopping patience of 3 epochs was used. Each setup was run 10 times. Table 1 reports the mean F_1 score for slot filling over the runs. Bidirectionality improved the baseline performance by approximately 3 points. Adding dropout achieved the best overall performance. The best-performing Adam configuration was also tested with SGD, which resulted in a slight performance drop. Table 2 shows the mean accuracy for intent classification over the runs. Incorporating bidirectionality and dropout into the baseline both improved accuracy. Similar to the slot filling task, experiments with the SGD optimizer resulted in a slight decline in performance compared to Adam. Figure 1 illustrates the convergence of the F_1 score over the training epochs. The addition of bidirectionality and dropout led to faster and more stable convergence of the F_1 metric. The experiments show that bidirectionality significantly improves performance in slot filling and intent classification tasks.

Second Part For the second part, the $\text{BERT}_{\text{BASE}}$ implementation was used as the baseline. All models were trained with a batch size of 64 for 10 epochs. For $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$, experiments with a 0.2 dropout rate were conducted.Table 3 shows the F_1 score for slot filling across the different models. The results indicate that smaller learning rates, such as 5×10^{-5}, are better suited for larger models like $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{BASE}}$, which fail to produce meaningful results at higher learning rates, such as 1×10^{-4}. Conversely, smaller BERT models perform better with higher learning rates, such as 1×10^{-4}. Although $\text{BERT}_{\text{LARGE}}$ with dropout achieves the best performance, it is worth mentioning that some smaller models, such as $\text{BERT}_{\text{MEDIUM}}$ and $\text{BERT}_{\text{MINI}}$, perform only slightly worse. Table 4 presents the accuracy for intent classification across the different models. Most conclusions drawn from Table 3 also apply to this task. The best learning rates are 1×10^{-4} and 5×10^{-5}, with larger models tending to perform better at smaller learning rates and smaller models working better at higher learning rates. $\text{BERT}_{\text{LARGE}}$ shows a significant performance drop at a learning rate of 1×10^{-4}. For intent classification, it is also worth noting that smaller models like $\text{BERT}_{\text{MEDIUM}}$ and $\text{BERT}_{\text{MINI}}$ perform only slightly worse than the best-performing model, $\text{BERT}_{\text{LARGE}}$ with dropout.

Table 1: *Mean Slot Filling F_1 scores in the first part*

Model	**Learning rate**			
Adam optimizer	**1e-4**	**5e-4**	**1e-3**	**5e-3**
LSTM	90.25*	91.24	91.04	91.13
LSTM + BiD	91.79	93.54	93.87	94.43
LSTM + BiD + DR0.2	92.58	93.90	94.39	**94.60**
SGD optimizer	**4**	**5**	**6**	**7**
LSTM + BiD + DR0.2	**94.07**	93.75	93.88	93.79

* This value was computed on only 7 runs, because 3 runs produced an error and therefore having an F_1 score of 0.
Note: Underlined values show the best F_1 score of a model and the bold value shows the best F_1 score over all models for a given optimizer.

Table 2: *Mean Intent Classification accuracy in the first part*

Model	**Learning rate**			
Adam optimizer	**1e-4**	**5e-4**	**1e-3**	**5e-3**
LSTM	92.68*	92.69	92.19	91.03
LSTM + BiD	94.82	95.70	96.10	95.50
LSTM + BiD + DR0.2	95.42	**96.15**	96.08	95.69
SGD optimizer	**4**	**5**	**6**	**7**
LSTM + BiD + DR0.2	95.40	95.60	95.50	**95.68**

* This value was computed on only 7 runs, because 3 runs produced an error and therefore having an intent accuracy of 0.
Note: Underlined values show the best classification accuracy of a model and the bold value shows the best classification accuracy over all models for a given optimizer.

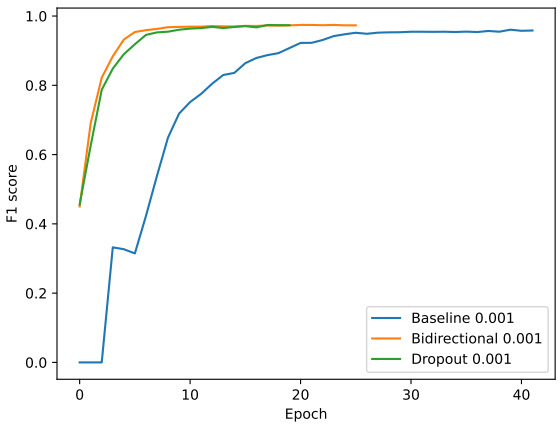

Figure 1: *The F_1 score convergence over the training epochs of selected models in the first part. The labels denote the model name with the learning rate*

Table 3: *Slot Filling F_1 scores in the second part*

Model	**Learning rate**		
Adam optimizer	**1e-5**	**5e-5**	**1e-4**
$\text{BERT}_{\text{LARGE}}$	92.73	95.55	0.00*
$\text{BERT}_{\text{LARGE}}$ + DR0.2	92.01	**95.66**	0.00*
$\text{BERT}_{\text{BASE}}$	90.74	95.23	94.71
$\text{BERT}_{\text{BASE}}$ + DR0.2	89.67	94.43	95.27
$\text{BERT}_{\text{MEDIUM}}$	85.51	94.41	94.78
$\text{BERT}_{\text{SMALL}}$	75.06	87.92	91.83
$\text{BERT}_{\text{MINI}}$	80.92	93.33	94.94
$\text{BERT}_{\text{TINY}}$	40.41	76.71	82.01

* The $\text{BERT}_{\text{LARGE}}$ does not produce meaningful slot filling results for this learning rate and therefore produces a F_1 score of 0.
Note: Underlined values show the best F_1 score of a model and the bold value shows the best F_1 score over all models.

Table 4: *Intent Classification accuracy in the second part*

Model	**Learning rate**		
Adam optimizer	**1e-5**	**5e-5**	**1e-4**
$\text{BERT}_{\text{LARGE}}$	97.31	97.42	70.77
$\text{BERT}_{\text{LARGE}}$ + DR0.2	97.42	**97.76**	70.77
$\text{BERT}_{\text{BASE}}$	94.96	97.31	**97.76**
$\text{BERT}_{\text{BASE}}$ + DR0.2	92.05	97.65	97.09
$\text{BERT}_{\text{MEDIUM}}$	93.17	96.53	97.65
$\text{BERT}_{\text{SMALL}}$	93.06	96.86	96.64
$\text{BERT}_{\text{MINI}}$	83.43	95.52	97.42
$\text{BERT}_{\text{TINY}}$	70.88	88.80	92.16

Note: Underlined values show the best classification accuracy of a model and the bold value shows the best classification accuracy over all models.

4. References

[1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[2] J. Schmidhuber, S. Hochreiter *et al.*, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[4] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," 2019. [Online]. Available: https://arxiv.org/abs/1908.08962

[5] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016. [Online]. Available: https://arxiv.org/abs/1609.08144

[6] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent

classification and slot filling," 2019. [Online]. Available: https://arxiv.org/abs/1902.10909

[7] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2047–2052.