NLU course project

Jie Chen (mat. 256177)

University of Trento

jie.chen-2@studenti.unitn.it

1. Introduction (approx. 100 words)

The objective of this project is to jointly improve the model performance on intent classification and slot filling. The baseline AIS model is improved by adding bidirectionality to the LSTM layer. In order to take into account both directions, the last two hidden layers are concatenated before passing them to the linear output layers. Dropout layers are then added to the hidden representations trying to enhance the performance. In the second part of the assignment, a BERT-based model is fine tuned trying to maximize the F1 score and accuracy. The dataset used is the ATIS dataset.

2. Implementation details (max approx. 200-300 words)

In the first part of the assignment, I introduced two boolean variables (`use_dropout` and `use_bidirectional`) allowing the model to be run with different parameter settings. I modified the ModelIAS architecture, which is based on a unidirectional LSTM, with a bidirectional LSTM. This change allows the model to process the input sequence in two directions, from lefttoright and from righttoleft in order to capture the context from both the past and the future in a sequence. Before passing the hidden layers to linear output layers, the output from both directions are concatenated. The bidirectionality benefits both the intention classification and slot filling. Intention classification typically uses a representation of the whole utterance while in slot filling, the meaning of a word often depends not only on the preceding word but also on the subsequent word. Then, a dropout layer is applied as the regularization technique which randomly sets to zero a fraction of neurons according to the dropout probability trying to help prevent overfitting. In this implementation, the dropout is applied to the last hidden layer of LSTM with a dropout value of 0.5.

In the second part of the project, the previous code is adapted to fine-tune a pretrained BERT model using a multitask leaning setting on intent classification and slot filling. To ensure the compatibility with BERT's input format, each utterance is tokenized word by word using the BERT tokenizer, following the approach described in the paper [1]. The special token [CLS] is added at the beginning of each utterance to serve as the sequence-level representation for intent classification, and [SEP] is added at the end to mark the sequence boundary. The BERT model requires the creation of three input tensors `input_ids`, `attention_masks`, `token_type_ids`. From the previous implementation, the slot2id and intent2id dictionaries are kept unchanged since they do not require tokenization while word2id dictionary is replaced by BERT tokenizer. In case a single word is split into multiple subtokens, the slots label is assigned to the first subtoken, while the remaining subtokens receive a special 'pad' label. This keeps the slot label sequence aligned with the tokenized input sequence length.

3. Results

First part. The table 2 below shows all the parameter settings I have experimented with and the corresponding results. Starting from the baseline IAS model architecture, I obtained the best performance with a bigger model. The additional modification of bidirectionality further improves the result, as the model can capture richer context. Applying dropout layers also enhances model performance by preventing overfitting and improving generalization. The evaluation metrics, F1 score and intention accuracy, increased correspondingly from 92.9% to 94.4% and from 94.2% to 95.0%, respectively.

Second part. The experiment is conducted in a single run. The fine-tuned pre-trained BERT-base [2] model achieves better performance with respect to the custom recurrent network architecture trained from scratch. In the table below 3 we can see the result obtained. I also experimented with different learning rates and dropout probabilities. Ultimately, the best results is achieved with a learning rate of 0.0001 and a dropout of 0.2, yielding an F1 score of 95.3% and an intention accuracy of 97.2%.

The table 1 below reports the best result obtained and the figure 1 shows the evolution of the loss during training for the best configuration.

4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[2] 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. [Online]. Available: https://arxiv.org/abs/1810.04805

Model	Slot F1	Intent Acc
IAS	0.929 ± 0.003	0.942 ± 0.002
IAS + BI	0.940 ± 0.002	0.945 ± 0.005
IAS + BI + DROP	**0.944 ± 0.003**	**0.950 ± 0.003**
BERT	**0.953**	**0.972**

Table 1: *Slot F1 and Intent Accuracy for IAS models with different modifications and fine-tuned BERT.*

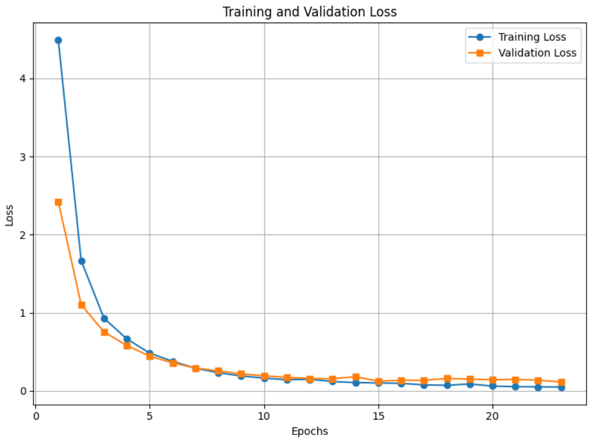

Figure 1: *Training and validation loss of BERT model*

Model	**hid_size**	**emb_size**	**lr**	**Slot F1**	**Intent Acc**
IAS	400	500	0.0001	0.929 ± 0.003	0.942 ± 0.002
IAS + BI	400	500	0.0001	0.940 ± 0.002	0.945 ± 0.005
IAS + BI + DROP	400	500	0.0001	0.944 ± 0.003	0.950 ± 0.003
IAS	200	300	0.0001	0.922 ± 0.002	0.935 ± 0.004
IAS + BI	200	300	0.0001	0.939 ± 0.003	0.946 ± 0.004
IAS + BI + DROP	200	300	0.0001	0.933 ± 0.002	0.947 ± 0.004

Table 2: *Slot F1 and Intent Accuracy for different IAS model configurations. BI = bidirectional, DROP = dropout applied.*

Model	**Dropout**	**lr**	**Slot F1**	**Intent Acc**
BERT	0.1	0.0002	0.949	0.970
BERT	0.1	0.0002	0.950	0.966
BERT	0.1	0.0001	0.952	0.968
BERT	0.2	0.0001	0.953	0.972

Table 3: *Slot F1 and Intent Accuracy for fine-tuned BERT models with different optimizers, learning rates, and dropout values.*