JOINT INTENT DETECTION AND SLOT FILLING BASED ON CONTINUAL LEARNING MODEL

Yanfei Hui, Jianzong Wang* , Ning Cheng, Fengying Yu, Tianbo Wu, Jing Xiao

Ping An Technology (Shenzhen) Co., Ltd.

ABSTRACT

Slot filling and intent detection have become a significant theme in the field of natural language understanding. Even though slot filling is intensively associated with intent detection, the characteristics of the information required for both tasks are different while most of those approaches may not fully aware of this problem. In addition, balancing the accuracy of two tasks effectively is an inevitable problem for the joint learning model. In this paper, a Continual Learning Interrelated Model (**CLIM**) is proposed to consider semantic information with different characteristics and balance the accuracy between intent detection and slot filling effectively. The experimental results show that CLIM achieves state-of-the-art performace on slot filling and intent detection on ATIS and Snips.

Index Terms— Slot Filling, Intent Detection, Continual Learning

1. INTRODUCTION

Natural Language Understanding (NLU) typically includes the intent detection and slot filling tasks, aiming to identify intent and extract semantic constituents from the utterance. Intent detection is essentially a classification problem and therefore popular classifiers like Linear Regression (LR), K-Nearest Neighbor (KNN), Support Vector Machines (SVMs) and deep neural network methods can be applied [1, 2]. Slot filling should be considered as a sequence labeling task that tags the input word sequence $x = (x_1, x_2, ..., x_T)$ with the slot label sequence $y = (y_1, y_2, ..., y_T)$. It should be noted that alignment is explicit in slot filling unlike machine translation and speech recognition, thus alignment-based models typically work well[3, 4]. Popular approaches to solving sequence labeling problems include Conditional Random Fields (CRFs), and Recurrent Neural Networks (RNNs) based approaches [5]. Particularly Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) models have achieved good performance.

Due to the accumulation of errors, pipeline methods usually fail to achieve satisfactory performance. Accordingly, some works suggested using one joint model for slot filling and intent detection to improve the performance via mutual

enhancement between two tasks. Recently, several joint learning methods for intent detection and slot filling were proposed to improve the performance over independent models [6, 7]. It has been proved that attention mechanism [8] can improve model's performance by dealing with long-range dependencies than currently used RNNs. So far attention-based joint learning methods have achieved the state-of-the-art performance for joint intent detection and slot filling [9, 10].

However, the prior work seems to lose sight of the fact that slot filling and intent detection are strongly correlative and the two tasks need encoding information with different characteristics. Among them, slot filling task needs more accurate encoding information. And there is a problem of alignment between slot and word, while intent detection only focuses on the overall semantic information, which is not sensitive to the word location information. Apart from that, there is a phenomenon of "seesaw" in the later stage of training the joint model, that is to say, when the accuracy of one task increases, it often causes the accuracy of the other task to decline.
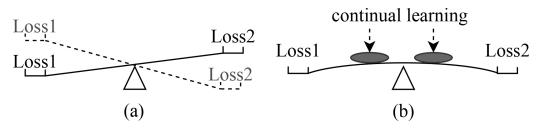

Fig. 1. Multitask joint training process

As shown in Figure 1a, when using the joint model to solve two tasks at the same time, the increase of task 1 accuracy will lead to the deterioration of task 2 performance, and vice versa. In this paper, CLIM is proposed to further enrich encoding information and take the cross-impact between two task into account. We introduce the idea of continuous learning into the training process, which makes the model perform satisfactorily in both tasks, as shown in Figure 1b.

In summary, the contributions of this paper are as follows:
- Applying the idea of continuous learning to solve the phenomenon of "precision seesaw" in the process of multi task training.
- Proposing a variant dual-encoder model structure to enrich the semantic information and location information contained by adopting different encoding forms.

---
*Corresponding author: Jianzong Wang, jzwang@188.com

2. PROPOSED APPROACH

2.1. Variant Dual-encoder Interrelated Model

We make a novel attempt to bridge the gap between word-level slot model and the sentence-level intent model via a dual-encoder model architecture, as shown in Figure 2.
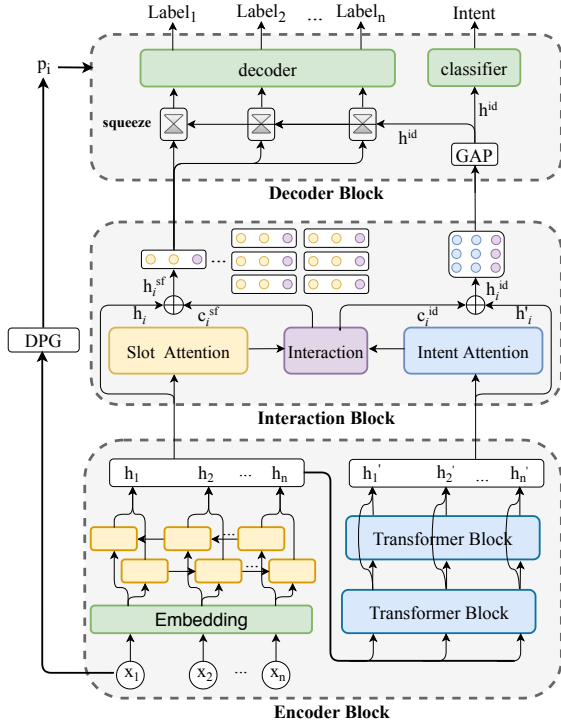

Fig. 2. Variant Dual-encoder Interrelated Model

2.1.1. Variant Dual-encoder

As mentioned above, CLIM apply two encoders (RNN and Transformer) to encode the sentence separately. As shown in Figure 2, there are two encoders in the Encoder Block, one on the left is a Bi-directional Long Short-Term Memory (BiLSTM), which can encode the input information in sequence. The additional encoder on the right is a bilayer transformer block [11]. Transformer models constitute the current state-of-the-art in machine translation and are becoming increasingly popular also in other domains [12, 13, 14], such as language modelling.

The BiLSTM reads in the input sentences $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ forward and backward, and generates two sequences of hidden states $\boldsymbol{h}_t^f$ and $\boldsymbol{h}_t^b$. A concatenation of $\boldsymbol{h}_t^f$ and $\boldsymbol{h}_t^b$ forms a final BiLSTM state $\boldsymbol{h}_t = concat[\boldsymbol{h}_t^f, \boldsymbol{h}_t^b]$ at time step t. On the other side, the hidden states generated by two transformer blocks are joined together by residual mechanism. In order to make the model have a faster inference speed, CLIM directly takes the output of LSTM as the input of the double-layer

transformer block, so as to get another different granularity encoding information. Then this encoder can also get a hidden state sequence $(\boldsymbol{h}_1', \boldsymbol{h}_2', ..., \boldsymbol{h}_n') : \boldsymbol{h}_i' = tanh(\boldsymbol{h}_i^1 + \boldsymbol{h}_i^2)$, where the hidden state $\boldsymbol{h}_i^1$ stand for the output of first transformer block, and the hidden state $\boldsymbol{h}_i^2$ represents the output of the second transformer block at time step i. The ith context vector $\boldsymbol{c}_i^{sf}$ in the slot attention matrix and $\boldsymbol{c}_i^{id}$ in the intent attention matrix are computed as the weighted sum of encoders' hidden states: $\boldsymbol{c}_i^{sf} = \sum_{j=1}^{T} \boldsymbol{\alpha}_{i,j} \boldsymbol{h}_j$ and $\boldsymbol{c}_i^{id} = \sum_{j=1}^{T} \boldsymbol{\alpha'}_{i,j} \boldsymbol{h}_j'$. The attention weight $\boldsymbol{\alpha}$ is acquired the same as that mentioned introduced in [9]. And the superscript 'sf' and 'id' represent slot filling and intent detection respectively. More formally, let $\boldsymbol{h}_j, \boldsymbol{h}_j' \in \boldsymbol{R}^{d \times s}$ be the jth sentence hidden states obtained by two encoders, respectively.

2.1.2. Interaction Modes between Encoders

Then, in the Interaction Block of CLIM, two attention matrices are used for information interaction by using this mechanism: $\boldsymbol{c}_i^{sf}, \boldsymbol{c}_i^{id} = \boldsymbol{c}_i^{id}, \boldsymbol{c}_i^{sf}$. The interaction mechanism can be defined in a variety of ways. We found that a straightforward implementation using matrix exchange can already achieve good results. In the Interaction Block, two kinds of hidden state vectors with different granularity are joined together and then sent to the Decoder Block to complete slot filling and intent detection tasks. The hidden state vector h^{sf} and h^{id} that is eventually fed to the Decoder Block can be obtained by concate function: $\boldsymbol{h}_i^{sf} = concat[\boldsymbol{h}_i, \boldsymbol{c}_i^{sf}]$ and $\boldsymbol{h}_i^{id} = concat[\boldsymbol{h}_i', \boldsymbol{c}_i^{id}]$. Then, the final hidden state used for intent detection can be obtained by global average pooling. As shown in Figure 2, the Decoding Layer includes a decoder and a classifier for slot filling and intent detection, respectively. Next we introduce the details of decoder and classifier.

2.1.3. Decoding Layer

For the classification task of intent detection, CLIM obtains the final intent category distribution by fusing hidden state $\boldsymbol{h}_i^{id}$. The classifier obtains the final intent in the following ways:

$$y_{intent} = softmax(W_{intent} \cdot GAP(\boldsymbol{h}_i^{id})) \qquad (1)$$

Besides, CLIM compressed the input information of decoder which can remove the information noise only related to the training set, so as to improve the generalization ability of the model. The compression process is realized by using one-dimensional convolution operation.

$$\boldsymbol{s}_i = conv1d(concat[\boldsymbol{h}_i^{sf}, \boldsymbol{h}^{id}]) \qquad (2)$$

where $\boldsymbol{s}_i$ is the input for decoder of slot filling. At last, the compressed dense representation will be used for sequence annotation.

2.2. Dynamic Parameter Generation

Several neural networks can be used to implement Dynamic Parameter Generation (DPG) for parameter generation, e.g., convolutional neural network (CNN), RNN and multilayer perceptron (MLP). The objective of this paper is not to explore all possible implementations. We found that a straightforward implementation using MLP can already achieve good results. Formally, DPG can be written as:

$$\boldsymbol{p}_i = f(\boldsymbol{z}_i; \mu) = \sigma(\boldsymbol{w}_D \boldsymbol{z}_i + \boldsymbol{b}) \tag{3}$$

where μ is the activation function, and $\boldsymbol{w}_D$ and b are the parameters of DPG and denoted by μ.

3. EXPERIMENTS AND RESULTS

3.1. Data

ATIS (Airline Travel Information Systems) data set contains audio recordings of people making flight reservations. As described in Table 2, ATIS data set contains 4978 train and 893 test spoken utterances. We also use Snips, which is collected from the Snips personal voice assistant. This data set contains 13084 train and 700 test utterances. We evaluate the model performance on slot filling using F1 score, and the performance on intent detection using classification error rate.

Table 1. Dataset details statistics

Dataset	ATIS	Snips
Vocab Size	722	11241
Average Sentence Length	11.28	9.05
Intents	21	7
Slots	120	72
Training Samples	4478	13084
Validation Samples	500	700
Test Samples	893	700

3.2. Training Procedure

LSTM cell is used as the basic RNN unit in the experiments. Given the size the data set, our proposed model set the number of units in LSTM cell as 200. Word embeddings of size 1024 are pre-trained and fine-tuned during mini-batch training with batch size of 20. Dropout rate 0.5 is applied behind the word embedding layer and between the fully connected layers. And our proposed model use Adam optimization method. In the process of model training, after a certain number of epochs, we tried to fix the weight of the word embedding layer so that the parameters would not be updated.

3.3. Joint Learning

Table 2 shows our joint training model performance on intent detection and slot filling comparing to previous reported results where the compared baselines include the state-of-the-art sequence-based joint model using bidirectional LSTM [7] and slot-gated model [16]. As shown in this table, our proposed model CLIM achieves **96.34**% F1 score on slot filling task and **98.43**% intent classification accuracy on ATIS dataset. Comparing with other models, it can be found that CLIM has the same accuracy as the optimal result in intent detection task and achieves state-of-the-art on slot filling task, even better than BERT, the pre-trained model with large parameters [10]. Moreover, on the Snips dataset, it can be found that although F1 score only slightly improves on slot filling tasks, our proposed model has made significant progress in the accuracy of intent detection. CLIM reduces the error rate of intent detection from **1.4**% to **0.84**%, and achieves **40.0**% reduction in error rate.

3.4. Continual Learning

It is found that if the traditional continuous learning method is used to train the model, the intent detection task will be learned first, and then the slot filling task will be learned, or vice versa [19]. However, through the previous experiments, we have found that the performance of solving two tasks at the same time by using the joint model is more than training on one task alone, so we need to modify the continuous learning method. Figure 3 and Figure 4 show the independent/joint/continual learning performance on intent detection and slot filling. We find CLIM loses 0.06% for slot filling on ATIS but wins joint learning 0.05% on Snips. We can observe that introducing more parameters does not always perform better. The different

Table 2. Performance of different joint models on two datasets.

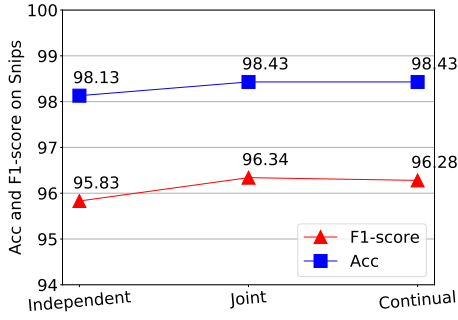| Model	ATIS		SNIPS	
	Slot(F1)	Intent(Acc)	Slot(F1)	Intent(Acc)
Atten. enc-dec NN with aligned inputs[9]	95.87	98.43	-	-
Atten.-BiRNN[9]	95.98	98.21	-	-
Enc-dec focus[15]	95.79	-	-	-
Slot-Gated[16]	95.2	94.1	88.8	97.0
CAPSULE-NLU[17]	95.2	95.0	91.8	97.3
SF-ID network(with CRF)[18]	95.75	97.76	91.43	97.43
Joint BERT[10]	96.1	97.5	97.0	98.6
CLIM	**96.34**	**98.43**	**97.03**	**99.00**

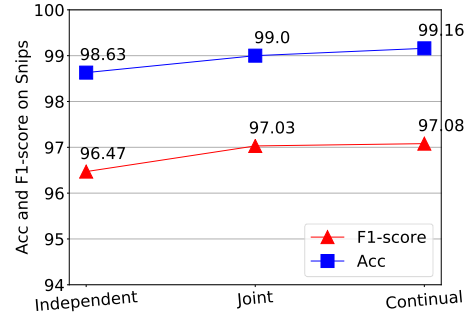**Fig. 3**. Performance of three methods on ATIS

Fig. 4. Performance of three methods on Snips

performance of continuous learning in two datasets should be due to the different size of vocabulary and types of slots. Meanwhile, CLIM wins joint learning 0.16% for intent detection on Snips and achieves the same result on ATIS. It makes sense since the classifier could use features from LSTM and Transformer simultaneously and different model structures complement each other for classification. In particular, our architecture performs best for both cases. In fact, DPG does not introduce new model structure or complicated operations and the number of parameters are almost the same.

3.5. Contrast Study

To further evaluate the advances of variant dual-encoder architecture for joint learning, we tested different dual-encoder structures. Note that all the variants are based on joint learning with information compression mechanism :

- BiLSTM-BiLSTM[B-B], where both encoders are bidirectional LSTM.
- BiLSTM-Transformer[B-T], where the left encoder is a bi-directional LSTM, and the other encoder is a double-layer transformer with residual connection.
- BiLSTM-Transformer(V)[B-T(V)], where the input of transformer is no longer word vector, but the encoding result of bidirectional LSTM. This is the structure adopted by CLIM model in this paper.
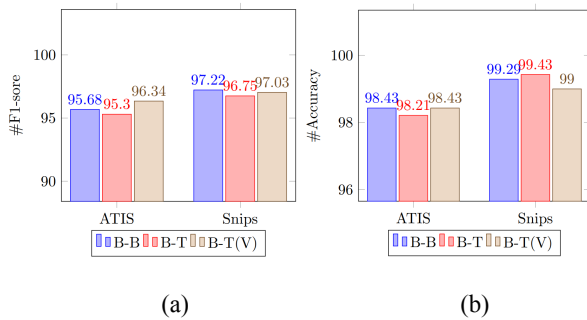

Fig. 5. Performance of different encoders on two datasets.

Figure 5 shows the joint learning performance of our model on ATIS data set and Snips data set by changing the structure of dual-encoder at a time. We found that the F1 score of slot filling on ATIS dataset decreased slightly with BiLSTM-BiLSTM, but the model can achieve the optimal accuracy in intent detection task. More importantly, it can achieve state-of-the-art for two tasks on Snips dataset at the same time. BiLSTM-Transformer does not perform well on ATIS dataset, and although it improves the accuracy of slot filling task in Snips dataset, the F1 score in slot filling task decreases a lot. Due to the small number of stack layers of transformer, it is difficult to play the encoding ability as powerful as BERT. BiLSTM-Transformer(V) can achieve state-of-the-art for two tasks on ATIS dataset at the same time. Although CLIM model has better results than other models on Snips dataset, the improvement of F1 score and accuracy is not as good as BiLSTM-BiLSTM. As shown in Figure 5, the model with BiLSTM-BiLSTM performs better than CLIM on Snips dataset, which is due to shorter sentences of Snips compared with ATIS dataset. As listed in the table, the variant dual-encoder model struture contributes to both slot filling and intent classification task.

4. CONCLUSION

In this paper, we proposed a continual learning model architecture to address the problem of accuracy imbalance in multitask. Using a joint model for the two NLU tasks simplifies the training process, as only one model needs to be trained and deployed. Our proposed model achieves state-of-the-art performance on the benchmark ATIS and SNIPS.

5. ACKNOWLEDGEMENTS

6. REFERENCES

[1] Ruhi Sarikaya, Geoffrey E. Hinton, and Bhuvana Ramabhadran, "Deep belief nets for natural language call-routing," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5680–5683, 2011.

[2] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, "Recurrent convolutional neural networks for text classification," in *American Association for Artificial Intelligence(AAAI)*, 2015.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," in *Conference and Workshop on Neural Information Processing Systems(NIPS)*, 2014.

[4] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[5] Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan ernocký, and Sanjeev Khudanpur, "Extensions of recurrent neural network language model," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5528–5531, 2011.

[6] Daniel Guo, Gökhan Tür, Wen tau Yih, and Geoffrey Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 554–559, 2014.

[7] Dilek Zeynep Hakkani, Gökhan Tür, Asli Çelikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *Conference of the International Speech Communication Association(Interspeech)*, 2016.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *Computing Research Repository(CoRR)*, vol. abs/1409.0473, 2014.

[9] Bing Liu and Ian Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *Conference of the International Speech Communication Association(Interspeech)*, pp. 685–689, 2016.

[10] Qian Chen, Zhu Zhuo, and Wen Wang, "Bert for joint intent classification and slot filling," *ArXiv*, vol. abs/1902.10909, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Conference and Workshop on Neural Information Processing Systems(NIPS)*, 2017.

[12] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Peng Shi, "Neural speech synthesis with transformer network," in *American Association for Artificial Intelligence(AAAI)*, 2018.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT)*, 2018.

[14] Wieting John, Neubig Graham, and Berg-Kirkpatrick Taylor, "A bilingual generative transformer for semantic sentence embedding," *Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pp. 1581–1594, 2020.

[15] Zhu Su and Yu Kai, "Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding," in *ICASSP 2017 - 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[16] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT)*, 2018.

[17] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu, "Joint slot filling and intent detection via capsule neural networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[18] E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[19] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *International Conference on Learning Representations(ICLR)*, 2019.