ASIM: Explicit Slot-Intent Mapping with Attention for Joint Multi-intent Detection and Slot Filling

Jingwen Chen[1], Xingyi Liu[2], Mingzhi Wang[1(✉)], Yongli Xiao[1], Dianhui Chu[1], Chunshan Li[1], and Feng Wang[2]

[1] Harbin Institute of Technology, Weihai, China
{chenjw,wangmz,xiaoyl,chudh,lics}@hit.edu.cn
[2] Weichai Power Co., Ltd., Weifang, China
{liuxingy,wangfeng05}@weichai.com

Abstract. The accurate analysis of a user's natural language statement, including their potential intentions and corresponding slot tags, is crucial for cognitive intelligence services. In real-world applications, a user's statement often contains multiple intentions, and most existing approaches either mainly focus on the single-intent research problems or utilizes an overall encoder directly to capture the relationship between intents and slot tags, which ignore the explicit slot-intent mapping relation. In this paper, we propose a novel Attention-based Slot-Intent Mapping Method (ASIM) for joint multi-intent detection and slot filling task. The ASIM model not only models the correlation among sequence tags while considering the mutual influence between two tasks but also maps specific intents to each semantic slot. The ASIM model can balance multi-intent knowledge to guide slot filling and further increase the interaction between the two tasks. Experimental results on the MixATIS dataset demonstrate that our ASIM model achieves substantial improvement and state-of-the-art performance.

Keywords: intent detection · slot filling · multi-intent · deep learning · attention

1 Introduction

The accurate analysis of the potential intentions in a user's natural language statement, as well as the corresponding slot tags that correspond to these intentions, is very important for cognitive intelligence services. Intent detection and slot filling are two core components of the cognitive service system. Intent detection aims to output the real intent of the user and solve the problem of what the user wants to do. Slot filling is to mark the important words in user input, which is to extract the details needed in service provision. Take the statement "Help me book a flight ticket from Beijing to New York." as an example, intention detection can be regarded as the text classification problem, which needs to output a user's real intention, i.e., "booking air tickets". Slot filling task can be

H. Zhang et al. (Eds.): NCAA 2023, CCIS 1870, pp. 215–228, 2023.
https://doi.org/10.1007/978-981-99-5847-4_16

thought of as a sequence labeling problem, requiring an output of a sequence, e.g., "O, O, O, O, O, B-from_location, O, B-to_location I-to_location".

In past research work, intent detection and slot-filling tasks were normally considered as independent tasks, but later researchers [2] considered that there was a correlation between them since the two tasks always appear together in the same conversation and have a mutual influence. Although many joint models have achieved good performance, these methods are based on the assumption that user utterance contains only simple single intent. However, in a real scenario, this is not the case. According to Gangadharaiah R et al. [2], 52% of user sentences in Amazon's customer service system involve multiple intents. Hence, in the process of real utterance, users will be faced with changing intent halfway or involving multiple intents in one sentence. For example, "Play jay Chou's latest single. No, just play the music video for the new song". In this case, the user suddenly changed his initial request. Or the user may have more than one demand, for example, "How much is the air ticket to Beijing during the National Day holiday, and tell me the recent weather in Beijing." Therefore, it is necessary to accurately identify all the intents in the user's utterance, which is very important work to provide information for the subsequent services.

In the early time, multi-intent detection is regarded as a text multi-label classification problem. However, the most multi-label classifier can work with long text, whereas the multi-intention detection task always works with short user utterances. Compared with single-intent detection, in a short text, there exist three main problems in multi-intent detection: 1. How to find out that users' utterances have multiple intents, and what is the difference between multi-intent utterances and single-intent utterances; 2. How to find out the number of intents hidden in the utterances after confirming that the utterance is multi-intent; 3. How to accurately identify all user intents.

In addition to the above problems, the multi-intent model still presents a unique challenge: how to effectively incorporate multiple intents knowledge to guide the slot prediction, since each word in a sentence has a different relevance for a different intent. Reference [3] proposed a slot gate mechanism, which flows intent information to the slot-filling task. Reference [4] proposed a new self-attention mechanism model, which enhanced the gate mechanism through intent information. The model used the intent information as the gate state information of the slot. Despite the promising performance, most of the previous works directly use multi-intent knowledge to predict the tags appearing of each slot word in a sentence, which would introduce part of noise information. Take the utterance "How much is the air ticket from New York to Beijing during the National Day holiday and tell me the recent weather in Beijing." for example (Fig. 1), if multiple intents information is directly used to guide slot filling of all words in a sentence, irrelevant information will be introduced and lead to poor performance. As shown in Fig. 1 (a), for the word "New" and "York", the intent "GetWeather" is almost irrelevant. Obviously, using the same intent knowledge to predict all slot tags may bring ambiguity. Therefore, for different words in one

sentence, how to introduce more detailed intent information is a crucial problem of intent detection and slot filling model.

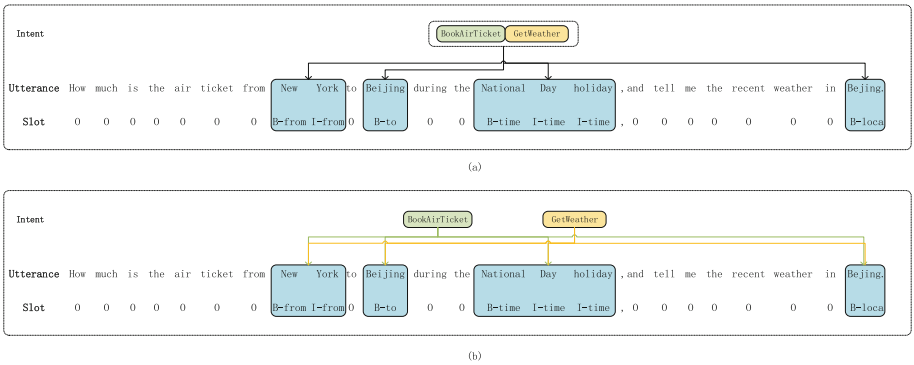

Fig. 1. Gangadharaiah et al. [2]: Utterance-level intent is shared by all the slots(a) vs ASIM: Word-level intent-slot mapping information is used by each slot(b)

In this paper, a novel Attention-based Slot-Intent Mapping Method (ASIM) is proposed, which not only can model the correlation among sequence tags while considering the mutual influence between two tasks, but also can map specific intent to each semantic slot. Unlike most existing models that implicitly share information between intent detection and slot filling through shared encoders, our ASIM model adopts respective encoders for intent detection and slot filling, which achieves the first information sharing by exchanging hidden state information between the two task encoders. More than that, our ASIM model constructs another information interaction in the decoder stage, where the importance coefficient of each word and intent information is calculated through the attention mechanism, which refines the guidance effect of multi-intent information on word slot filling.

We conclude the main contributions of this paper:

- the proposed ASIM model will explore the multi-intent detection task and slot filling task together, which can construct two information-sharing mechanisms in the encoder and decoder module to capture the correlation between intent detection and slot filling as well as analyze users' slot-intent mapping on the word level.
- We use an attention mechanism to balance the degree of closeness between multiple intents and words to guide slot filling in the decoder stage.
- we conducted experiments on MixATIS data-set to validate our hypothesis and the results illustrate that our ASIM model achieved better performance than other intent detection model.

The rest of the paper is organized as follows. In Sect. 2, several related literature will be introduced including intent detection, slot filling, as well as joint

model. Section 3 mainly discuss more details about ASIM model. In Sect. 4, we conduct the experiments to verify the performance of ASIM model from different perspectives. Finally, we conclude our work and give the further plan.

2 Related Works

In this section, several related literature will be introduced including intent detection, slot filling, as well as joint model.

2.1 Intent Detection Tasks

Intent detection is always seen as a text classification problem. Therefore, most of the traditional classification methods can be used for intent detection, including Naive Bayes model [8], support vector machine(SVM) [9] and logistic regression [10]. Traditional methods can be divided into rule-based template semantic recognition methods [11] and statistics-based classification algorithms [12]. Even without a lot of training data, the rule-based template methods still achieve good results. However, the template needs to be formulated by experts, and the template reconstruction requires a lot of economic cost and time cost to adopt this method. The classification algorithms based on statistics need to extract the key information of corpus, so these methods need a lot of training data. Therefore, traditional intent detection methods cannot meet higher requirements. With the great success of artificial neural networks in other fields, intent detection methods based on deep neural networks have become popular. With the success of convolutional neural network (CNN) in the field of computer vision, researchers [13] employ CNN network to determinate the 5-gram features of sentence, and maximum pooling was applied to generate the feature embedding vector of words. As in [14], Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) are applied to intent detection according to the sequential nature of user utterances.

2.2 Slot Filling Tasks

Slot filling task can be formulated as a sequence labeling problem. The previous methods to solve the slot filling problem are mainly divided into three categories.

1) Dictionary approach [15]**.** This method searches for dictionary keywords mainly through string matching. Since a large number of corpus is needed to construct the dataset, this method consumes manpower and faces the problem of data scarcity.

2) Rule-based approach [16–18]**.** This method marks keywords in user utterance by rule matching. Because domain experts are required to make rules, so the costs are high. In addition, scalability is poor. With the gradual increase of user's requirements, experts are needed to constantly improve the existing rules, and rule conflicts are easy to occur.

3) Traditional machine learning method [19–22]**.** This method takes artificially labeled corpus as the training set and optimizes model parameters through multiple training to minimize the target loss function. Not only a large number of labeled training data are required, but also features are manually constructed.

With the high-speed development of deep neural network [23,24]. many AI algorithms have also been applied to slot filling, such as recurrent neural network (RNN), convolutional neural network (CNN) and various combinations of traditional machine learning methods [25].

2.3 Joint Model

Considering intent-slot relation and information-sharing mechanism between intent and slot tasks, the researchers began to train the two tasks together. The joint model is not only take advantage of information interaction between two tasks, but also simplifies the training process by training only one model. The early research literature in this field is the CNN+Tri-CRF method [13], which utilizes CNN networks as a shared encoder to integrate the intent detection and slot filling tasks, and then employs a CRF layer to handle dependencies among slot tags. Guo et al. [27] proposed a joint training methods of the recursive neural network (RecNNs) for intent detection and slot filling tasks. Zhang et al. [28,29] employ a Gated recurrent unit (GRU) to learn the representation of each time step in RNN and predict the label of each slot tags. Liu et al. [1] proposed introducing attention to the alignment-based RNN models which can add additional information to the intent detection and slot filling tasks. Goo ea al. [3] utilize a slot gate structure to learn the relationship between intent and slot attention vectors and achieve better semantic segment results by the global optimization. Wang et al. [5] employ a Bi-model based RNN network structures to handle the cross-impact between the intent detection and slot filling tasks. The key points of Bi-model are two inter-connected bidirectional LSTMs structure and two different cost functions in an asynchronous training. Qin et al. [26,30] propose two attention mechanism-based models that adopt Stack Propagation which can directly employ the intention embedding as input for slot filling, and capture the semantic information of intent. In recent years, pre-trained language models [31–33] have significantly enhanced the performance of many natural language processing (NLP) applications. Chen ta al. [34] investigates BERT pre-trained model to address the poor generalization capability on intent detection and slot filling. Zhang et al. [35] design a effective encoder-decoder framework to improve the performance of intent detection and slot filling tasks.

3 Methodology

In this part, We will begin by defining the joint intent detection and slot filling tasks. Then, we will give the detail of the Attention-based Slot-Intent Mapping model (ASIM), which calculates the correlation between multiple intents and

the current word, and then uses the information of multiple intents to guide slot filling. The model we proposed, increases the mutual influence between two tasks.

3.1 Problem Definition

The input sequence is defined as $x = (x_1, x_2, x_3, ...x_n)$. Intent detection is treated as a classification problem, and final output is intent label $y^1 = (y_1^1, y_2^1, y_3^1, ...y_m^1)$, where m is the number of the intents the input sequence contains. Slot filling is treated as a sequence labeling problem, and the final output is $y^2 = (y_1^2, y_2^2, y_3^2, ...y_n^2)$.

3.2 ASIM Model

Figure 2 illustrates the network structure of the ASIM model, in which intent detection and slot filling use different encoders and decoders respectively. The left part of the network is designed for intent detection and the right part is designed for slot filling. In the bottom part, the two encoders read and encode the input sentence. Then the encoded information is passed to the decoder for outputting the predicted intents and slot tags. The subsequent sentences give the details of how the encoders and decoders operate for intent detection and slot filling.

3.3 Encoder

BiLSTM. Considering a specific relationship between intent detection and slot filling, most studies use shared encoders to share information between intent detection and slot filling tasks. However, these approaches are not only poorly interpretable but also not obvious for intent detection and slot filling information flow. Hence, to explicitly describe the interaction between intent detection and slot filling, the proposed ASIM uses two encoders corresponding to intent detection and slot filling, respectively.

BiLSTM consists of two LSTM units. For the input sequence $x = (x_1, x_2, x_3, ...x_n)$, BiLSTM obtains the forward hidden state vector $\overrightarrow{h^i} = (\overrightarrow{h_1^i}, \overrightarrow{h_2^i}, ..., \overrightarrow{h_n^i})$ from x_1 to x_n, and obtains the backward hidden state vector $\overleftarrow{h^i} = (\overleftarrow{h_1^i}, \overleftarrow{h_2^i}, ..., \overleftarrow{h_n^i})$ from x_n to x_1. The final hidden state vector $H^i = (h_1^i, h_2^i, ..., h_n^i)$ is obtained by concatenating forward hidden state vector and backward hidden state vector, where $i = 1$ corresponds to the task of intent detection and $i = 2$ corresponds to slot filling.

Attention Mechanism. Generally, sentences with multiple intentions are longer than those with a single intent. As the length of text increases, although BiLSTM can capture information from both sides of the sentence, it still causes some information loss. In addition, the correlation between the current tag and

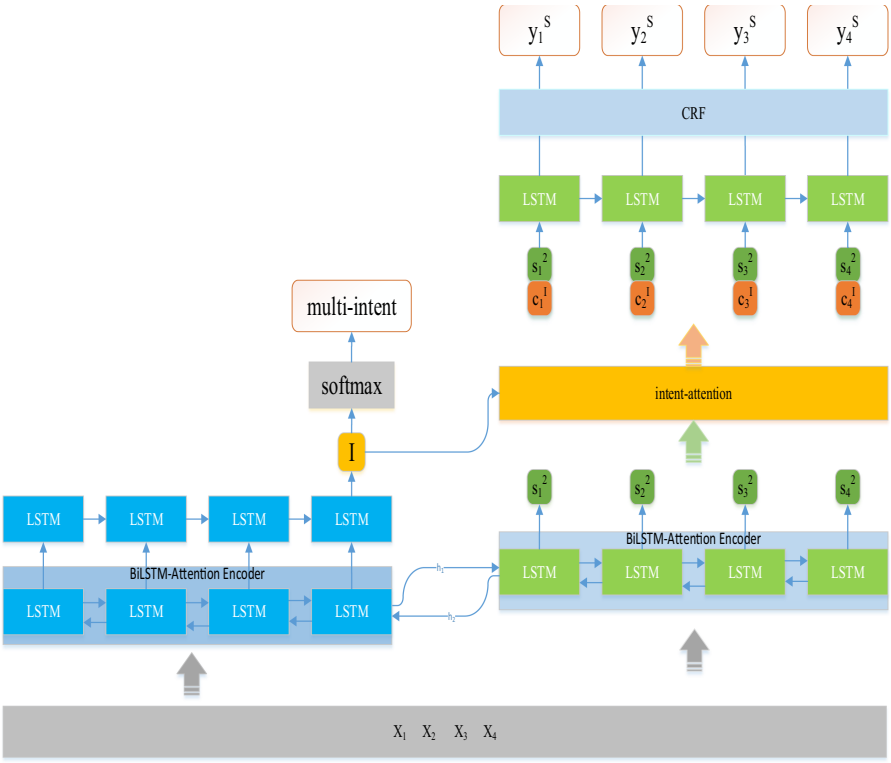

Fig. 2. The structure of the ASIM model.

the other tags in the sentence is not the same. Therefore, a self-attention mechanism is added in the encoder stage of slot filling in this paper, aiming to assign different importance degrees to other words related to the current word when encoding the current word information. The attention mechanism not only makes up for the information loss generated by BiLSTM but also obtains the correlation information between the current tag and the other tags in one sentence. The following is the introduction of the attention mechanism.

For each hidden state h_i^m, the context vector c_i^m is obtained by calculating the weighted sum of the hidden states:

$$c_i^m = \sum_{j=1}^{n} a_{i,j}^m h_j^m \tag{1}$$

The attention score can be obtained from the following formula:

$$a_{i,j}^m = \frac{exp(e_{i,j}^m)}{\sum_{k=1}^{n} exp(e_{i,k}^m)} \tag{2}$$

$$e_{i,j}^m = g(s_{i-1}^m, h_k^m) \tag{3}$$

where g is the feedforward neural network and where $m = 1$ corresponds to intent detection and $m = 2$ corresponds to slot filling.

We concatenate these two representations as the final encoding representation:

$$E = [H|C] \tag{4}$$

where H is the final hidden state matrix and C is the context matrix.

3.4 Decoder

Intent Detection Decoder. Multi-intent detection is regarded as a multi-label classification problem. In order to carry out the explicit, hidden layer state information interaction between the intent detection and slot filling, the intent decoder receives a hidden state of the slot encoder and carries out the information sharing between the intent detection and the slot filling. The hidden state of the intent decoder at time i is shown as follows:

$$s_i^1 = \phi(s_{i-1}^1, h_{i-1}^1, h_{i-1}^2, c_i^1) \tag{5}$$

$$y_{intent}^1 = \sigma(\hat{y}_i^1 | s_{i-1}^1, h_{i-1}^1, h_{i-1}^2, c_i^1) \tag{6}$$

where $y_{intent}^1 = \{y_{intent,1}^1, y_{intent,2}^1, ..., y_{intent,N_I}^1\}$ is the intent output of the sentence, N_I is the number of intent of the current sentence, and σ is the activation function.

Our model refers to paper [6] to output all user intents through a hyperparameter t_u. Suppose the prediction intent is $I = (I_1, I_2, I_3, ...I_n)$, I_i represents $y_{I_i}^I$ greater than t_u, where t_u is the hyperparameter obtained by fine-tuning the validation data set. For example, if $y^I = \{0.9, 0.3, 0.6, 0.7, 0.2\}$ and $t_{0.5}$, then we can get $I = (1, 3, 4)$.

Slot Filling Decoder. The intent encoder hidden state h_{i-1}^1 and the slot encoder hidden state h_{i-1}^2 are utilized for slot filling:

$$s_i^2 = \varphi(h_{i-1}^2, h_{i-1}^1, s_{i-1}^2, c_i^2) \tag{7}$$

$$y_i^2 = \sigma(\hat{y}_n^2 | h_{i-1}^2, h_{i-1}^1, s_{i-1}^2, c_i^2) \tag{8}$$

where σ is the activation function and the s_{i-1}^2 is the hidden state of slot decoder.

Slot-Intent Mapping with Attention. The core of this part is to balance the degree of closeness between multi-intent and the slot and use the balanced multi-intent information to guide the slot filling. The concrete implementation is as follows.

Firstly, according to the current word and the predicted multiple intents, we calculate the degree of closeness between each intent and the current word and get the score of each intent, which is used as the attention score of the currently hidden unit:

$$a_{i,j} = \frac{exp(e_{i,j}^I)}{\sum_{k=1}^{m} exp(exp(e_{i,k}^I))} \tag{9}$$

$$e_{i,j}^{I} = g(I_{i,j}, h_k^2) \tag{10}$$

where I is the predicted multiple intents. h_i^2 is the current slot hidden state. The calculated weight a_i^I is the weight of intent, which represents the degree of closeness of the corresponding intent to the current word.

By summing up all the weighted predicted intents, the context vector of intents c_i^I to the current word is obtained:

$$c_i^I = \sum_{j=1}^{m} a_{i,j}^I I_{i,j} \tag{11}$$

where c_i^I represents the integration of all the information related to the current sentence intent, which is used to guide the slot filling. The output of the slot filling decoder is:

$$s_i^2 = \varphi(e_i^2, h_{i-1}^2, h_{i-1}^1, s_{i-1}^2, c_i^I) \tag{12}$$

$$y^2 = \sigma(\hat{y}_i^2 | e_i^2, h_{i-1}^2, h_{i-1}^1, s_{i-1}^2, c_i^I) \tag{13}$$

where σ is the activation function.

CRF. If the model without CRF is used for slot filling, the slot label with the highest score of each label is selected as the slot label of the word. However, in practical applications, the tag with the highest score may not always be the most suitable one. In order to solve this issue, a CRF layer is added after the slot filling decoder.

The CRF layer will model several dependencies of the slot tags to ensure that the predicted slot is more suitable so as to increase the accuracy of correct slot prediction. These constraints can be learned automatically through the CRF layer during data training.

For sentence 1 "please give me the flight times the morning on united airline for september twentieth from philadelphia to san francisco". The true tag of the phrase "flight times" is "$B-flight\_time$ $I-flight\_time$", but the slot tag predicted by the model without CRF is "O $I-flight\_time$". For sentence 2 "what type of ground transportation is available at philadelphia airport and then how many first class flights does united have today". The true tag of the phrase "philadelphia airport" is "$B-airport\_name$ $I-airport\_name$", but the slot tag predicted by the model without CRF is "$B-city\_name$ $I-airport\_name$". The model with CRF can reduce the errors mentioned in these two sentences in most cases.

The CRF layer can learn some constraints for correctly predicting slots. For BIO-tagged data, the possible constraints are:

1) Instead of "$I-X$", an X element should begin with "$B-X$" or "O". For example, tag "$I-flight\_time$" in sentence 1 should not be the beginning of the "flight_time" element. Thus, the model that add CRF layers can correctly predict "$B-flight\_time$" as the beginning of the element "flight_time".
2) For slot label sequence "$B-label_1$ $I-label_2$ $I-label_3$...", $label_1$, $label_2$ and $label_3$ should be the same entity category. As shown in sentence 2, "$B-city\_name$ $I-airport\_name$" is clearly wrong.

3.5 Asynchronous Training

We employ two different cost functions to train the ASIM model with an asynchronous fashion. We define the loss function of intention network is $\mathcal{L}_1$, and the loss function of slot filling networks is $\mathcal{L}_2$. $\mathcal{L}_1$ and $\mathcal{L}_2$ are formulated as:

$$\mathcal{L}_1 \triangleq -\sum_{i=1}^{k} \hat{y}_{intent}^{1,i} log(y_{intent}^{1,i}) \tag{14}$$

and

$$\mathcal{L}_2 \triangleq -\sum_{j=1}^{n}\sum_{i=1}^{m} \hat{y}_{j}^{2,i} log(y_{j}^{2,i}) \tag{15}$$

where k denotes the number of intent label types, m represents the number of semantic tag types, n is the length of a word sequence.

4 Experimental Results

4.1 The Data-Set Description

To assess the efficiency of the proposed ASIM model, experiments are carried out on MixATIS with multiple intents. Due to the scarcity of multi-intent data sets, reference [6] constructed multi-intent data set MixATIS data on commonly used single-intent data set ATIS. ATIS has 656 words, 18 intents, and 130 slot labels. By using conjunctions "and" to combine sentences with various intentions, a sentence can have one to three intentions, in which the proportion of each number of intents is [0.3,0.5,0.2]. The number of training sets, verification sets, and test sets of the final MixATIS data set is 18000, 1000 and 1000, respectively.

4.2 Baselines

1) Attention BiRNN. Liu et al. [2] propose introducing attention to RNN model and bring additional information to the intent detection and slot filling tasks.

2) Slot-Gated Atten. Goo et al. [3] use a slot-gated based RecNNS to explicitly consider the information-sharing between the two tasks.

3) Bi-Model. Wang et al. [4] employ two inter-connected bidirectional LSTMs structure and two different cost functions to improve model performance.

4) SF-ID Network. Niu et al. [6] design bi-directional interrelated network to model direct correlation between intent detection and slot filling.

4.3 The Experiment Design

This paper deals with the data set MixATIS as in [7]. The identifier "UNK" represents those that occur in the test data but not in the training data, and the number is represented by a string of varying lengths "DIGIT" based on its digits.

The ASIM model uses deep learning PyTorch framework for training. In the training stage, the word feature dimension d_C is 300, the maximum sentence length is 130, and the BiLSTM unit dimension d is 200. The threshold value $t_u = 2$.

4.4 The Experiment Results

We first employ the MixATIS benchmark datasets to show the performance of the ASIM model. The specific results are shown in Table 1. Compared with the previous benchmark models, the model proposed in this paper improves slot F_1 and Intent acc on MixATIS data set and intent detection has a big improvement. It can be seen from the results that the Intent Acc is 1.6% higher than that of Bi-Model. It can be analyzed that the probable reason is that the attention mechanism we added in the encoder captures the important sentence information so that the content of sentence important information contained in the embedding vector is increased. Besides, the slot and intent mapping module also play a positive role in improving the Intent Acc. Since intent detection and slot filling are related to each other, the improvement of one task can also have a positive influence on the other.

Table 1. Comparison of experimental results

Model	MixATIS	
	Slot (F1)	Intent (Acc)
Attention BiRNN	86.6	71.6
Slot-Gated	**88.1**	65.7
Slot-gated Inten	86.7	66.2
Bi-Model	85.5	72.3
SF-ID	87.7	63.7
the ASIM model	**87.19**	**73.90**

Ablation Experiments. Model modification. Table 2 shows the ablation experiment results. As can be seen from Table 2, both the attention mechanism of intent detection and slot filling added in the encoder and the Slot-Intent mapping module have a positive effect on the experimental results. The attention mechanism of the encoder part improves the two tasks. Compared with the Bi-model, the slot $F1$ score is improved by 2.21, and the Intent Acc is improved

by 1.1% in Bi-Model(with encoder Attention). The reason is that the attention mechanism added in the encoder captures the information of the important words in the sentence and reduces the information loss caused by the LSTM model. Therefore, the decoder can receive input vectors containing more information about those important words. The slot-Intent Mapping module does not clearly improve slot filling but improves intent detection by 0.3% in Bi-Model(with slot-intent Mapping). The possible reason for this phenomenon is the Slot-Intent Mapping module significantly increases the interaction between intent detection and slot filling, which not only provides guidance to each other but also potentially introduces a bit of error information. But in general, the model proposed in this paper achieves good results in intent detection and slot filling.

Table 2. Comparison of ablation results

Model	MixATIS	
	Slot (F1)	Intent (Acc)
Bi-Model	85.50	72.30
Bi-Model (with encoder Attention)	**87.71**	73.40
Bi-Model (with slot-intent Mapping)	85.50	72.60
the ASIM model	**87.19**	**73.90**

5 Conclusions

In this paper, A novel attention-based slot-intent mapping (ASIM) model was proposed for joint multi-intent detection and slot filling tasks. The ASIM model not only can model the correlation among sequence tags while considering the mutual influence between two tasks, but also can map specific intention to each semantic tag. In particular, this ASIM uses two encoding structure to achieve more obvious information interaction between the two tasks and uses an attention mechanism to balance the degree of closeness between multiple intents and words to guide slot filling in the decoder stage. Then, the interaction between the two tasks is mutually reinforcing. A CRF layer is added after the slot filling decoder which can model several dependencies of the slot tags to ensure that the predicted labels are more suitable. After experimental verification on a real-world dataset, the ASIM model has achieved the best performance than other state-of-art methods.

References

1. Liu, B., Lane, I.: Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling (2016)
2. Gangadharaiah, R., Narayanaswamy, B.: Joint multiple intent detection and slot labeling for goal-oriented dialog. In: Proceedings of the 2019 Conference of the North (2019)
3. Goo, C.-W., et al.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of NAACL (2018)
4. Li, C., Li, L., Qi, J.: A selfattentive model with gate mechanism for spoken language understanding. In: Proceedings of EMNLP (2018)
5. Wang, Y., Shen, Y., Jin, H.: A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In: Proceedings of NAACL (2018)
6. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of ACL (2019)
7. Qin, L., Xu, X., Che, W., Liu, T.: AGIF: an adaptive graph-interactive framework for joint multiple intent detection and slot filling. In: EMNLP Findings (2020)
8. Mccallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization, pp. 41–48 (1998)
9. Haffner, P., Tur, G., Wright, J.H.: Optimizing SVMs for complex call classification. In: IEEE International Conference on Acoustics, pp. 632–635 (2003)
10. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale Bayesian logistic regression for text categorization. Technometrics **49**(3), 291–304 (2007)
11. Dowding, J., et al.: Gemini: a natural language system for spoken-language understanding (1994)
12. Pengju, Y.: Research on Natural Language Understanding in Conversational Systems. Tsinghua University, Beijing (2002)
13. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 78–83 (2013)
14. Ravuri, S., Stolcke, A.: Recurrent neural network and LSTM models for lexical utterance classification. In: Interspeech (2015)
15. Wang, Q.: Biological named entity recognition combining dictionary and machine learning. Dalian University of Technology (2009)
16. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
17. Cucerzan, S., Yarowsky, D.: Language independent named entity recognition combining morphological and contextual evidence. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 90–99 (1999)
18. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 1–8 (1999)
19. Wei, L., Mccallum, A.: Rapid development of Hindi named entity recognition using conditional random fields and feature induction. ACM Trans. Asian Lang. Inf. Process. **2**(3), 290–294 (2003)

20. Bikel, D.M., Miller, S., Schwartz, R., et al.: Nymble: a high-performance learning name-finder. Anlp, pp. 194–201 (1998)
21. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An algorithm that learns what's in a name. Mach. Learn. **34**(1–3), 211–231 (1999)
22. Borthwick, A., Grishman, R.: A maximum entropy approach to named entity recognition. Graduate School of Arts and Science. New York University (1999)
23. Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., Yu, D.: Recurrent neural networks for language understanding. In: Interspeech (2013)
24. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrentneural-network architectures and learning methods for spoken language understanding. In: Interspeech (2013)
25. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: SLT (2014)
26. Qin, L., Che, W., Li, Y., et al.: A stack-propagation framework with token-level intent detection for spoken language understanding (2019)
27. Guo, D., Tur, G., Yih, W.-T., Zweig, G.: Joint semantic utterance classification and slot filling with recursive neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 554–559 (2014)
28. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: IJCAI, vol. 16, pp. 2993–2999 (2016)
29. Liu, B., Lane, I.: Joint online spoken language understanding and language modeling with recurrent neural networks. arXiv preprint arXiv:1609.01462 (2016)
30. Qin, L., Ni, M., Zhang, Y., Che, W.: Cosda-ml: multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. arXiv preprint arXiv:2006.06402 (2020)
31. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1) (2019)
32. Sun, Y., et al.: Ernie 2.0: a continual pre-training framework for language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8968–8975 (2020)
33. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
34. Chen, Q., Zhuo, Z., Wang, W.: BERT for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909 (2019)
35. Zhang, Z., Zhang, Z., Chen, H., Zhang, Z.: A joint learning framework with BERT for spoken language understanding. IEEE Access **7**, 168:849–168:858 (2019)