Fan JF, Wang ML, Li CL *et al.* Intent-slot correlation modeling for joint intent prediction and slot filling. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(2): 309–319 Mar. 2022. DOI 10.1007/s11390-020-0326-4

Intent-Slot Correlation Modeling for Joint Intent Prediction and Slot Filling

Jun-Feng Fan (樊骏锋), Mei-Ling Wang (汪美玲), Chang-Liang Li* (李长亮), *Senior Member*, *CCF*
Zi-Qiang Zhu (朱自强), and Lu Mao (毛　璐)

AI Laboratory, KingSoft Corporation, Beijing 100190, China

E-mail: {fanjunfeng, wangmeiling1, lichangliang, zhuziqiang, maolu}@kingsoft.com

Received January 21, 2020; accepted September 20, 2020.

Abstract Slot filling and intent prediction are basic tasks in capturing semantic frame of human utterances. Slots and intent have strong correlation for semantic frame parsing. For each utterance, a specific intent type is generally determined with the indication information of words having slot tags (called as slot words), and in reverse the intent type decides that words of certain categories should be used to fill as slots. However, the Intent-Slot correlation is rarely modeled explicitly in existing studies, and hence may be not fully exploited. In this paper, we model Intent-Slot correlation explicitly and propose a new framework for joint intent prediction and slot filling. Firstly, we explore the effects of slot words on intent by differentiating them from the other words, and we recognize slot words by solving a sequence labeling task with the bi-directional long short-term memory (BiLSTM) model. Then, slot recognition information is introduced into attention-based intent prediction and slot filling to improve semantic results. In addition, we integrate the Slot-Gated mechanism into slot filling to model dependency of slots on intent. Finally, we obtain slot recognition, intent prediction and slot filling by training with joint optimization. Experimental results on the benchmark Air-line Travel Information System (ATIS) and Snips datasets show that our Intent-Slot correlation model achieves state-of-the-art semantic frame performance with a lightweight structure.

Keywords spoken language understanding, slot filling, intent prediction, Intent-Slot correlation, slot recognition

1 Introduction

Spoken language understanding (SLU) systems, which aim to capture semantic frame of human utterances, play a crucial role in spoken dialogue systems. SLU typically involves two basic tasks: intent prediction and slot filling [1]. Intent prediction identifies speakers' intent and slot filling extracts semantic constituents as constraints for natural language queries. Let us take a flight-related sentence as an example, "List flights from Boston to Baltimore on Friday", as shown in Fig.1(a). There is a specific intent type for the whole sentence and are different slot labels for each word.

Intent prediction and slot filling are generally treated as a classification task and a sequence labeling task respectively, and the two tasks are usually processed separately. Popular approaches for intent prediction include support vector machine (SVM) [2] and deep neural network methods [3, 4]. Different sequence labeling methods, such as conditional random fields (CRF) [5] and recurrent neural network (RNN) [6], have been applied to slot filling. To address the error propagation problems of independent models, serials of joint models for intent prediction and slot filling have been proposed [7–10]. Typical structures of neural joint models include encoder-decoder (or sequence to sequence) [8, 11], attention [11] and Bi-model RNN [10]. With joint models, performance can be improved via mutual enhancement between intent prediction and slot filling tasks [11–13].

Slots and intent have strong correlation for semantic frame parsing. Firstly, intents depend on the information of slots. A specific intent type is generally determined based on the sentence constituents, which give

Regular Paper
*Corresponding Author

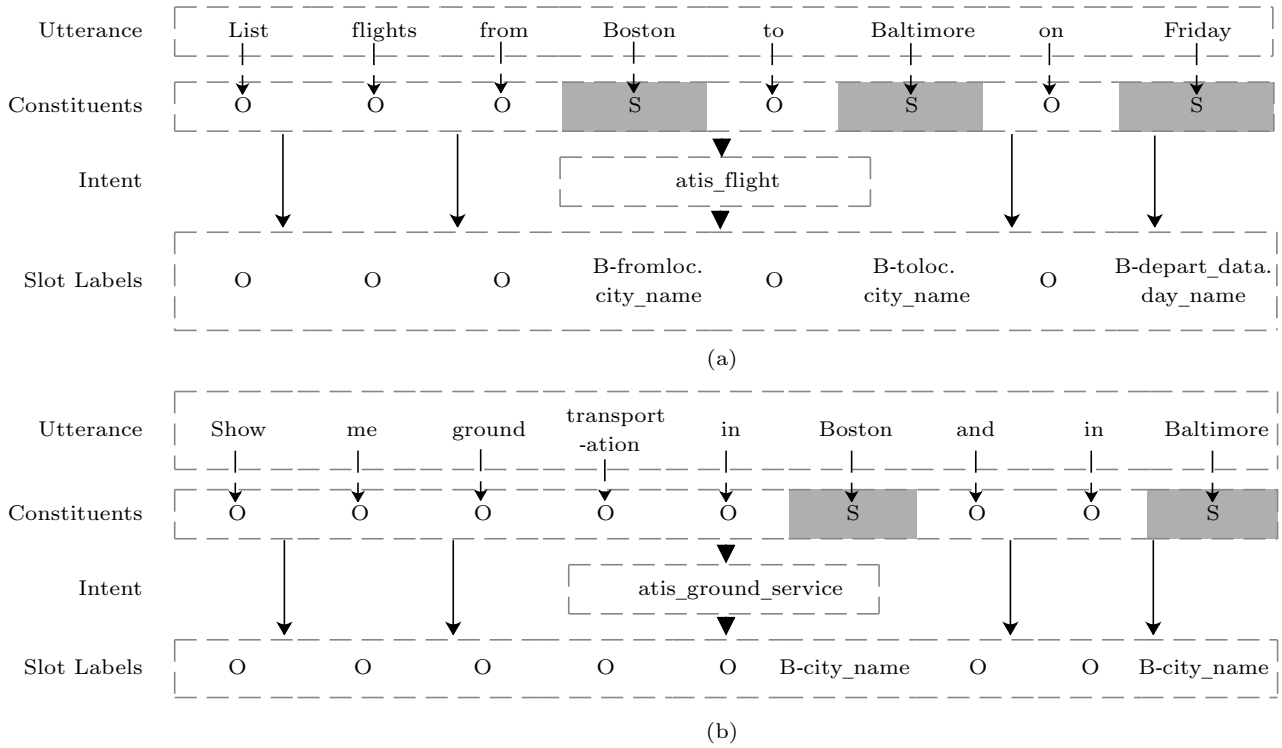

Fig.1. Two example utterances of Intent-Slot correlation, where constituent elements are tagged in the "SO" (Slot, Other) format and slots are tagged in the "BIO" (Begin, Inside, Other) format.

rich indication information to intent prediction. The words having slot tags (called as slot words) are the most typical prediction indicators. As shown in Fig.1,

1) in Fig.1(a), intent "atis_flight" is mainly determined based on "flights from Boston to Baltimore on Friday", where "Boston" and "Baltimore" describe locations, "Friday" describes depart date, and the other words denote auxiliary information about "flights";

2) in Fig.1(b), intent "atis_ground_service" is mainly determined based on "ground transportation in Boston and in Baltimore", where "Boston" and "Baltimore" describe locations, and the other words denote related auxiliary information.

Secondly, the intent type decides that words of certain categories should be used to fill as slots. For the examples in Fig.1(a) and Fig.1(b), "Boston" and "Baltimore" are recognized as location constraints and are tagged with proper slot labels corresponding to intent "atis_flight" and "atis_ground_service" respectively.

However, the Intent-Slot correlation is rarely modeled explicitly in existing studies, and hence may not be fully exploited. Most existing studies model Intent-Slot correlation implicitly by applying joint loss function or sharing internal information (such as hidden states) among multi-models [8, 10, 11, 14]. Recently, researchers have modeled the dependency of slots on intent explicitly by introducing a Slot-Gated mechanism [12] or by proposing a novel self-attentive model with gate mechanism [15], while they ignore the dependency of intents on slots. Zhang *et al.* [16] proposed a capsule-based neural network model with a dynamic routing schema to exploit hierarchical relationship among words, slots, and intents, and model the impact of the intent on slot indirectly by updating word vectors. E *et al.* [17] modeled the bi-directional interrelated connections for intent and slot explicitly by introducing a complex SF-ID network.

In this paper, we focus on modeling Intent-Slot correlation explicitly and propose a new lightweight framework for joint intent prediction and slot filling. Firstly we explore the effects of slot words on intent by differentiating them from the other words. As shown in Fig.1, slot words are tagged as "S" and the other words are tagged as "O". We propose to recognize slot words by solving a sequence labeling task with the bi-directional long short-term memory (BiLSTM) model [18]. Secondly, we improve attention-based intent prediction by leveraging information about differences between slot words and the other words. To improve slot filling, we leverage beforehand slot location information from the slot recognition layer and mean-

while integrate Slot-Gated mechanism [12] for modeling dependency of slots on intent. Finally, we obtain slot recognition, intent prediction and slot filling by training with joint optimization. We conduct experiments on the benchmark ATIS and Snips datasets and compare our model with sequence-based model [8], attention-based model [11], Slot-Gated model [12], capsule-based model [16] and SF-ID network [17] on semantic frame parsing related metrics. Experimental results show that leveraging our Intent-Slot correlation model achieves state-of-the-art semantic frame performance with a lightweight structure. In addition, we conduct an ablation study and a case study to evaluate our model.

The contributions of this paper are three-fold.

1) We explore the effects of slot words on intent and model with a sequence labeling task. In our knowledge, this is the first work that explicitly models the effects of slots on intent by sequence labeling.

2) Our design of slot recognition benefits intent prediction, slot filling and semantic frame parsing effectively.

3) We propose a new framework for joint intent prediction and slot filling by integrating slot recognition and Slot-Gated mechanism, which is lightweight and achieves state-of-the-art semantic frame performance.

2 Related Work

As slot filling and intent prediction can be treated as a sequence labeling task and an utterance classification task respectively, pipelined approaches are the initial choices and the two tasks are usually processed separately [2–6]. Different sequence labeling methods, such as conditional random fields (CRFs) [5] and recurrent neural network (RNN) [6], have been applied to slot filling. Popular approaches for intent prediction include support vector machine (SVM) [2] and deep neural network methods [3, 4]. However, pipelined approaches usually suffer from error propagation problems due to the independency of models [12].

To address the issues of pipelined approaches, serials of joint models for intent prediction and slot filling have been proposed [7–10] and performance has been improved via mutual enhancement between the two tasks [11–13]. Nowadays, structures of neural joint models typically include encoder-decoder (or sequence to sequence) [8, 11], attention [11] and Bi-model RNN [10]. Hakkani-Tür *et al.* [8] proposed an RNN-LSTM architecture for joint modeling of slot filling, intent determination and domain classification by building a joint multi-domain model, which enables multi-task deep learning and reinforces the data from multi-domain. Liu and Lane [11] explored joint models for intent prediction and slot filling with an encoder-decoder model and attention mechanism. With respect to the encoder-decoder model, they used one encoder and two decoders, where the first decoder generates sequential semantic tags and the second decoder generates intent types. They proposed another approach that consolidates hidden states information from an RNN slot filling model and then generates its intent type using an attention model. Wang *et al.* [10] designed new Bi-model based RNN network structures for joint intent prediction and slot filling by considering their cross-impact to each other. In their structures, two inter-correlated BiLSTMs are used for intent prediction and slot filling respectively, and an asynchronous training approach is designed to adapt to the new structures. In all these studies, Intent-Slot correlation is modeled implicitly by applying joint loss function or sharing internal information (such as hidden states) among multi-models.

With respect to studies on modeling of Intent-Slot correlation explicitly, Goo *et al.* [12] modeled the dependency of slots on intent by introducing a Slot-Gated mechanism, which leverages intent context vector for modeling slot-intent relationships in order to improve slot filling performance. Li *et al.* [15] proposed a novel self-attentive model with gate mechanism and utilized intent semantic representation for labeling slot tags. However, the dependency of intents on slots is ignored by them. Zhang *et al.* [16] proposed a capsule-based neural network model with dynamic routing schema to achieve synergistic effects for joint slot filling and intent detection, and they modeled the impact of intent on slot indirectly by updating word vectors. E *et al.* [17] modeled the connections for intent and slot explicitly by introducing an complex SF-ID network, in which they designed a new iteration mechanism to enhance the bi-directional interrelated connections. In this paper, we focus on modeling Intent-Slot correlation explicitly with a lightweight model.

3 Proposed Model

This section explains our Intent-Slot correlation model for joint intent prediction and slot filling. The model architecture is illustrated in Fig.2. As shown in Fig.2, the shared layer encodes an input word sequence by applying a BiLSTM model [18], and the encoding results are fed as input into the layers of slot recognition,

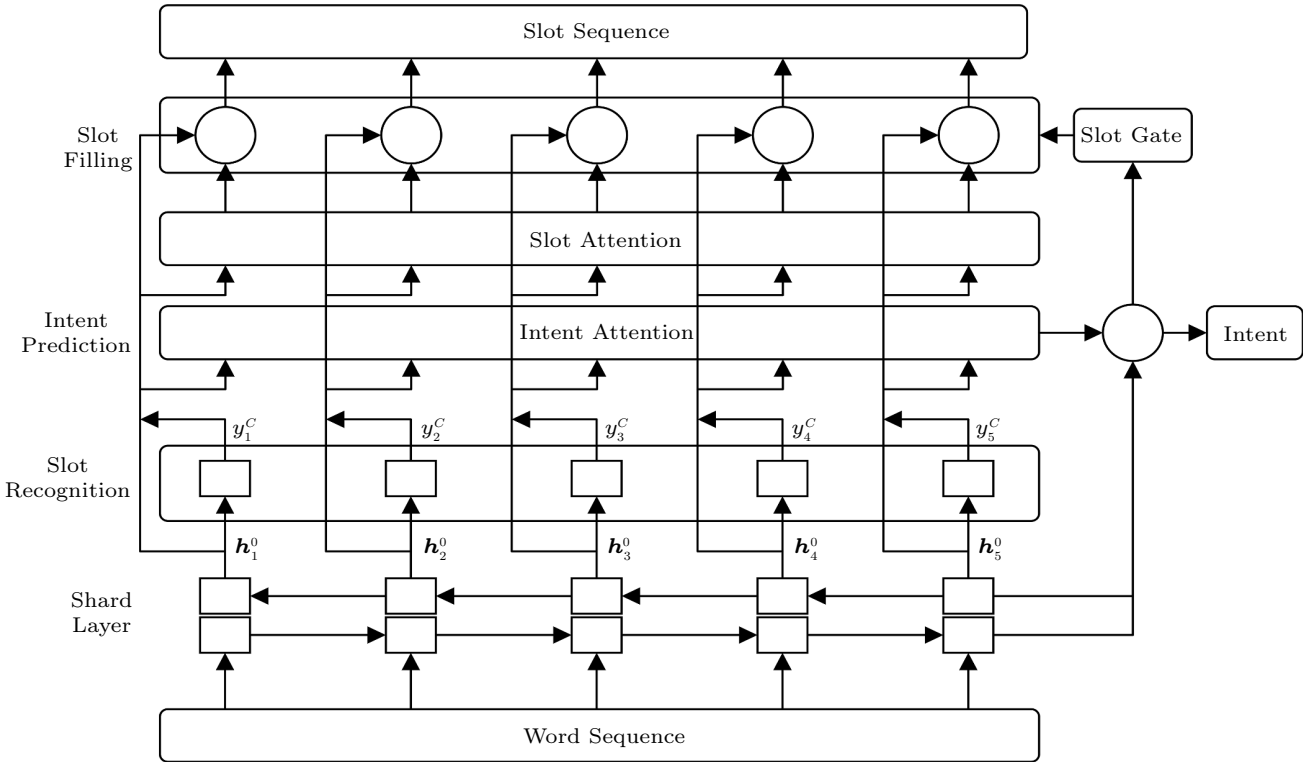

Fig.2. Architecture of Intent-Slot correlation model.

intent prediction, and slot filling. The slot recognition layer labels the word sequence as slot words and the other words, and the labeling results are fed as input into the layers of intent prediction and slot filling. With word encoding and slot recognition information, the intent prediction layer applies attention mechanism to decode intent types, and the slot filling layer applies the attention mechanism and Slot-Gated mechanism [12] to decode slot labels.

We first describe slot recognition, which models the impact of slots on intent in Subsection 3.1. Second, we describe intent prediction and slot filling based on the information provided by slot recognition in Subsections 3.2 and 3.3 respectively. Finally, in Subsection 3.4 we describe the joint optimization of slot recognition, intent prediction and slot filling.

3.1 Slot Recognition

We first focus on modeling the effects of slots on intent. The roles played by slot words and the other words are significantly different in intent prediction, where slot words generally describe crucial semantic information and constraints and the other words denote auxiliary semantic information. For intent "atis_flight" in Fig.1(a), "Boston" and "Baltimore" describe locations about "flights" and the other words denote auxiliary information about "flights". Therefore we explore the effects of slot words by differentiating them from the other words.

We divide constituent elements of each sentence into two categories, slot words tagged with "S" (Slot) and the other words tagged with "O" (Other). The "BIO" (Begin, Inside, Other) tagging schema for slot filling can be transformed into the "SO" schema by substituting "B" and "I" with "S". Our aim is to recognize slot words of sentences. Note that we do not recognize slot categories because wrong slot categories introduce noise to intent prediction. For the examples in Fig.1(a) and Fig.1(b), "Boston" and "Baltimore" are tagged with different slot categories in the two sentences as their intent types are different.

We first apply a BiLSTM model [18] on a shared layer to sequentially encode the sequence of T words $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ into hidden state sequence $\boldsymbol{h}^0 = (\boldsymbol{h}_1^0, \boldsymbol{h}_2^0, ..., \boldsymbol{h}_T^0)$, where the i-th hidden state $\boldsymbol{h}_i^0 = \left[\overrightarrow{\boldsymbol{h}_i^0}, \overleftarrow{\boldsymbol{h}_i^0}\right]$, $\overrightarrow{\boldsymbol{h}_i^0}$ is forward hidden state generated with forward LSTM and $\overleftarrow{\boldsymbol{h}_i^0}$ is backward hidden state generated with backward LSTM.

Then we regard slot recognition as a sequence labeling task that maps hidden state sequence $\boldsymbol{h}^0 =$

$(\boldsymbol{h}_1^0, \boldsymbol{h}_2^0, ..., \boldsymbol{h}_T^0)$ to its corresponding constituent label sequence $\boldsymbol{y}^C = (y_1^C, y_2^C, ..., y_T^C)$. We solve the sequence labeling task by first applying a BiLSTM model, which sequentially encodes hidden state sequence $\boldsymbol{h}^0$ into hidden state sequence $\boldsymbol{h}^1 = (\boldsymbol{h}_1^1, \boldsymbol{h}_2^1, ..., \boldsymbol{h}_T^1)$, where the i-th hidden state $\boldsymbol{h}_i^1 = \left[\overrightarrow{\boldsymbol{h}_i^1}, \overleftarrow{\boldsymbol{h}_i^1}\right]$. Then the hidden state sequence $\boldsymbol{h}^1$ is utilized to generate constituent labels:

$$y_i^C = \text{softmax}\left(\boldsymbol{W}^C \boldsymbol{h}_i^1 + \boldsymbol{b}^C\right),$$

where y_i^C is the constituent label of the i-th word in the sentence, $\boldsymbol{W}^C$ is the weight matrix, and $\boldsymbol{b}^C$ is the bias vector.

For the example in Fig.1(a), the shared layer takes sentence "List fights from Boston to Baltimore on Friday" as an input, and encodes corresponding word sequence into the hidden state sequence. Then the slot recognition component generates labels for each word, i.e., "S" for "Boston", "Baltimore" and "Friday", and "O" for "List", "fights", "from", "to", and "on".

3.2 Intent Prediction

We improve intent prediction with information about differences between slot words and the other words of slot recognition output.

Firstly, features $\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_T)$ are computed with hidden states $\boldsymbol{h}^0$ and slot recognition output $\boldsymbol{y}^C$:

$$\boldsymbol{f}_i = \boldsymbol{W}^F([\boldsymbol{h}_i^0 ; \boldsymbol{y}_i^C]),$$

where $\boldsymbol{W}^F$ is the weight matrix guaranteeing that $\boldsymbol{f}_i$ has the same shape with $\boldsymbol{h}_i^0$.

For each feature $\boldsymbol{f}_i$, we compute the intent context vector $\boldsymbol{c}_i^I$ as the weighted sum of features, $\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_T$, by the learned attention weights $\alpha_{i,j}^I$:

$$\boldsymbol{c}_i^I = \sum_{j=1}^{T} \alpha_{i,j}^I \boldsymbol{f}_j,$$

where the intent attention weights are computed as:

$$\alpha_{i,j}^I = \frac{\exp\left(e_{i,j}\right)}{\sum_{k=1}^{T} \exp\left(e_{i,k}\right)},$$
$$e_{i,k} = \sigma\left(\boldsymbol{f}_i \boldsymbol{W}_e^I \boldsymbol{f}_k\right),$$

where $\boldsymbol{W}_e^I$ is the weight matrix and σ is the activation function. $\boldsymbol{c}^I$ is computed as the sum of intent context vectors, $\boldsymbol{c}_1^I, \boldsymbol{c}_2^I, ..., \boldsymbol{c}_T^I$, to provide additional information to intent classification:

$$\boldsymbol{c}^I = \sum_{j=1}^{T} \boldsymbol{c}_j^I.$$

Then intent is computed with $\boldsymbol{h}_T^0$ and intent context vector $\boldsymbol{c}^I$:

$$y^I = \text{softmax}\left(\boldsymbol{W}_y^I \left(\boldsymbol{h}_T^0 + \boldsymbol{c}^I\right)\right),$$

where y^I is the intent type and $\boldsymbol{W}_y^I$ is the weight matrix.

For the example in Fig.1(a), intent prediction component computes intent attention based on feature vectors combining slot recognition outputs with hidden states, and predicts the intent of the input sentence, i.e., "atis_flight".

3.3 Slot Filling

We leverage beforehand slot locations information of slot recognition output to improve slot filling, and meanwhile integrate the Slot-Gated mechanism [12] for dependency of slots on intent. The Slot-Gated mechanism is proposed to use the information of intent context vector to model the impact of intent on slot, which can improve slot filling.

We first compute slot context $\boldsymbol{c}^S = (\boldsymbol{c}_1^S, \boldsymbol{c}_2^S, ..., \boldsymbol{c}_T^S)$ on features $\boldsymbol{f}$, where $\boldsymbol{c}_i^S$ is computed in the same manner as $\boldsymbol{c}_i^I$, and then we take as input $\boldsymbol{c}^S$ and $\boldsymbol{c}^I$ into the slot gate as illustrated in Fig.3 to get a weighted feature g of the joint context vector:

$$g = \sum_{i=1}^{T} \boldsymbol{v} \cdot \tanh\left(\boldsymbol{W}\boldsymbol{c}^I + \boldsymbol{c}_i^S\right),$$

where $\boldsymbol{v}$ and $\boldsymbol{W}$ are trainable parameters. In detail, g indicates the degree that the slot context and the intent context pay attention to the same part of the input sequence, and a larger g infers that the correlation between the slot and the intent is stronger and the intent context contributes the prediction result more reliably.

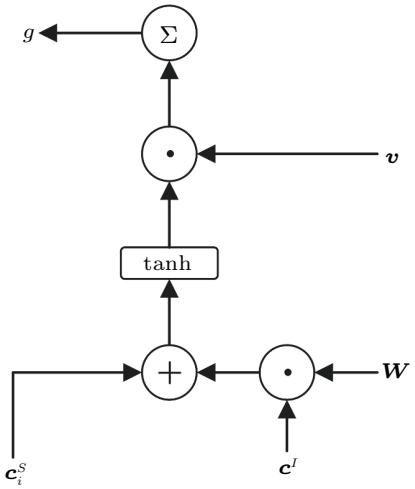

Fig.3. Illustration of slot gate [12].

With features $\boldsymbol{f}$, slot context $\boldsymbol{c}^S$ and g, the slot filling is modeled as below:

$$y_i^S = \text{softmax}\left(\boldsymbol{W}_y^S\left(g\boldsymbol{c}_i^S + \boldsymbol{f}_i\right)\right),$$

where y_i^S is the slot label of the i-th word and $\boldsymbol{W}_y^S$ is the weight matrix. Larger g indicates that intent context vector contributes more information for prediction of slots. For the example in Fig.1(a), slot attention is computed based on feature vectors combining slot recognition outputs with hidden states, then the slot gate gets a weighted feature with intent attention and slot attention, and finally the slot filling component generates labels for each word, i.e., "B-fromloc.city_name" for "Boston", "B-toloc.city_name" for "Baltimore", "B-depart_data.day_name" for "Friday", and "O" for "List", "flights", "from", "to", and "on".

As shown in [12], to compare the power of the slot gate in our model architecture, we also propose a model with only intent attention, where g and y_i^S are defined as follows:

$$g = \sum_{i=1}^{T} \boldsymbol{v} \cdot \tanh\left(\boldsymbol{W}\boldsymbol{c}^I + \boldsymbol{f}_i\right),$$
$$y_i^S = \text{softmax}\left(\boldsymbol{W}_y^S\left(g\boldsymbol{f}_i + \boldsymbol{f}_i\right)\right).$$

3.4 Optimization

We obtain slot recognition, intent prediction and slot filling jointly and formulate the objective as joint optimization:

$$\begin{aligned} & p\left(\boldsymbol{y}^C, y^I, \boldsymbol{y}^S|\boldsymbol{x}\right) \\ = & \prod_{n=1}^{T} p\left(y_n^C|\boldsymbol{x}\right) p\left(y^I|\boldsymbol{x}\right) \prod_{n=1}^{T} p\left(y_n^S|\boldsymbol{x}\right) \\ = & \prod_{n=1}^{T} p\left(y_n^C|x_1, \cdots, x_T\right) p\left(y^I|x_1, \cdots, x_T\right) \\ & \prod_{n=1}^{T} p\left(y_n^S|x_1, \cdots, x_T\right), \end{aligned}$$

where $p\left(\boldsymbol{y}^C, y^I, \boldsymbol{y}^S|\boldsymbol{x}\right)$ is the conditional probability of slot recognition, intent prediction, and slot filling given the input word sequence.

4 Experiment

To evaluate our model, we conduct experiments on the benchmark ATIS (air-line travel information System) [19] and Snips [20] datasets.

The ATIS dataset contains audio recording of people making flight reservations and is widely used in the SLU research. Selected from ATIS-2 and ATIS-3 corpora①, 4 978 utterances are divided into 4 478 utterances in the training set and 500 utterances in the development set. Selected from ATIS-3 NOV93 and DEC94 corpora, 893 utterances are contained in the testing set. There are 21 intent types and 120 slot labels in the training set.

The Snips dataset② is collected from the Snips personal voice assistant. There are 13 084 utterances in the training set and 700 utterances in the testing set. Another 700 utterances are contained in the development set. There are seven intent types and 72 slot labels in the training set.

Comparison between the ATIS and Snips datasets is shown in Table 1. The ATIS dataset is single-domain and has intents all about flight information with similar vocabularies, while the Snips dataset has more diverse intents and a larger vocabulary.

Table 1. Statistics of ATIS and Snips Datasets

	ATIS	Snips
Vocabulary size	722	11 241
Training set size	4 478	13 084
Development set size	500	700
Testing set size	893	700
Types of slots	120	72
Types of intents	21	7

Based on the benchmark ATIS and Snips datasets, we evaluate the SLU performance of our model using $F1$-score of slot filling, the accuracy of intent prediction, and the semantic frame accuracy of sentences.

In Subsections 4.1–4.3, we first compare our model with existing sequence-based model [8], the attention-based model [11], the Slot-Gated model [12], the SF-ID model [17] and the capsule-based model [16], then we conduct an ablation study of our model, and finally we conduct a case study of our model.

4.1 Model Comparison

4.1.1 Baselines

The compared baselines for joint intent prediction and slot filling include the followings:

①https://github.com/Microsoft/CNTK/tree/master/Examples/LanguageUnderstanding/ATIS/Data, Mar. 2022.
②https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines, Aug. 2020.

1) the sequence-based joint model using BiLSTM [8], which proposes an RNN-LSTM architecture for joint modeling of slot filling, intent determination, and domain classification by building a joint multi-domain model;

2) the attention-based model [11], which proposes joint models for intent prediction and slot filling with the encoder-decoder model and the attention mechanism;

3) the Slot-Gated model [12], which models the dependency of slots on intent by introducing a Slot-Gated mechanism and leverages the intent context vector for modeling slot-intent relationships;

4) the SF-ID network [17], which establishes direct connections for intent prediction and slot filling with SF subnet and ID subnet, where the SF-First model (referred to as SF-ID Network (SF-First)) executes the SF subnet first and the ID-First model (referred to as SF-ID Network (ID-First)) executes the ID subnet first;

5) the capsule-based model (referred to as CAPSULE-NLU) [16], which accomplishes slot filling and intent prediction via a dynamic routing-by-agreement schema and promotes slot filling with the inferred intent representation.

4.1.2 Experimental Setting

Under the same experimental settings as [17], we compare our model with the sequence-based model [8], the attention-based model [11], the Slot-Gated model [12] and the SF-ID model [17]. For each BiLSTM in the model, one layer is contained, and the hidden size is set to 64. The word embeddings size is set to 64. The batch size is set to 16 and the dropout rate to 0.01. The optimizer is adam.

We compare our Intent-Slot model with the capsule-based model [16] under a different experimental setting. In detail, the Intent-Slot model uses a BiLSTM with two layers. The batch size is set to 64. Both the word embedding size and the BiLSTM hidden size are set to 128. Under this experimental setting, our Intent-Slot model still has far fewer parameters than the capsule-based model.

4.1.3 Experimental Results

Table 2 displays the evaluation results of the Intent-Slot model compared with baseline models.

1) Compared with the sequence-based model [8], the attention-based model [11], the Slot-Gated model [12] and the capsule-based model [16], our Intent-Slot model achieves the best performance on both datasets. Firstly, both intent prediction and slot filling obtain significant improvement, demonstrating that our slot recognition module can benefit intent prediction and slot filling effectively. Secondly, semantic frame parsing for whole utterances obtains significant improvement, demonstrating that our Intent-Slot correlation model can benefit SLU effectively. Finally, our Intent-Slot model has far fewer parameters than the capsule-based model, demonstrating that our model performs better with a more lightweight structure.

2) Compared with the SF-ID model [17], our Intent-Slot model achieves close intent prediction accuracy, slot filling $F1$, sentence-level semantic frame accuracy, which means our Intent-Slot model achieves state-of-the-art semantic frame performance.

In sum, the experiments show that our Intent-Slot correlation model can achieve state-of-the-art semantic frame performance with a lightweight structure. This result is probably because that our slot recognition differentiates slot words from the other words to enhance intent prediction, and meanwhile slot locations in slot

Table 2. Model Comparison Results of Intent-Slot

Model	ATIS			Snips		
	Slot ($F1$)	Intent (Acc)	Sentence (Acc)	Slot ($F1$)	Intent (Acc)	Sentence (Acc)
Joint Seq [8]	94.3	92.6	80.7	87.3	96.9	73.2
Attention-Based [11]	94.2	91.1	78.9	87.8	96.7	74.1
Slot-Gated [12]	95.4	95.4	83.7	89.3	96.9	76.4
SF-ID Network (SF-First) [17]	**95.6**	**97.4**	86.0	90.3	97.3	78.4
SF-ID Network (ID-First) [17]	**95.6**	96.6	86.0	**90.5**	97.0	78.4
Intent-Slot	**95.6**	96.5	**86.4**	90.0	**97.4**	**78.5**
CAPSULE-NLU [16]	95.2	95.0	83.4	91.8	97.3	80.9
Intent-Slot*	**95.8**	**95.3**	**85.5**	**91.9**	**97.4**	**81.7**

Note: $*$ denotes the Intent-Slot model is under different experimental settings, in which the embedding size and the hidden size are set to 128 (see details in Subsection 4.1.2). Acc: Accuracy. The best results of all models and better results between Intent-Slot models are in bold.

recognition output improve slot filling, and the joint optimization for the three tasks achieves better semantic frame performance.

4.2 Ablation Study

We first conduct an ablation study to evaluate whether and how each part of our model contributes to our full model. In detail, we ablate four important components and conduct experiments under the same setting as [17].

1) *w/o* (*without*) *Slot Recognition, Where No Slot Recognition Is Contained in Our Intent-Slot Model*. No slot labeling results are fed as input into the layers of intent prediction and slot filling, and the model is the same as the Slot-Gated model with full attention.

2) *w/o Intent Attention, Where No Attention Mechanism Is Performed in the Intent Prediction Layer*. The intent prediction layer decodes intent types with word encoding and slot recognition information.

3) *w/o Slot Attention, Where No Attention Mechanism Is Performed in the Slot Filling Layer*. The slot filling layer applies the Slot-Gated mechanism to decode slot labels with word encoding and slot recognition information.

4) *w/o Slot Gated, Where No Slot-Gated Mechanism Is Performed in the Slot Filling Layer*. The slot filling layer applies attention mechanism to decode slot labels with word encoding and slot recognition information.

Table 3 shows the performance of our model on the ATIS and Snips datasets by removing one module at a time. We find that if we remove the slot recognition layer from the holistic model, the semantic frame performance drops dramatically. The result can be interpreted as that our slot recognition can provide helpful information for the global optimization of joint intent prediction and slot filling. We can see that slot recognition does improve performance a lot in a large scale. If we remove the Slot-Gated mechanism from the holistic model, the semantic frame performance drops a lot, which demonstrates that the Slot-Gated mechanism is essential to determine the slot labels. If we remove slot attention or intent attention from the holistic model, the performance variation differs on the ATIS and Snips datasets, and it is probably because the two datasets have different complexity.

We then evaluate the effects of training epochs on $F1$-score of slot recognition, $F1$-score of slot filling, accuracy of intent prediction, and semantic frame accuracy of sentences. Fig.4 displays the evaluation results.

1) The average $F1$-score of slot recognition is 99.3 on ATIS and 98.7 on Snips respectively, which are strong guarantees for improving $F1$-score of slot filling, the accuracy of intent prediction, and the semantic frame accuracy of sentences.

2) When epochs increase, $F1$-score of slot filling, the accuracy of intent prediction, and the semantic frame accuracy of sentences increase to a high level (as shown with dotted lines) at the beginning and then fluctuate continuously around peak values.

3) Overall, $F1$-score of slot filling and the accuracy of intent prediction reach peak values earlier than the semantic frame accuracy of sentences, as our model needs some more epochs to fine-turn with respect to the semantic frame accuracy of sentences based on Intent-Slot correlation.

4.3 Case Study

Table 4 displays four examples for case study. Case 1 and case 2 show two examples of the ATIS dataset for comparing the Intent-Slot model and the Slot-Gated model. Case 1 shows that the Intent-Slot model recognizes the label of "ap58" correctly with the slot location information from slot recognition, but the Slot-Gated model cannot. Case 2 illustrates that with the same slot information, the Intent-Slot model predicts the intent type correctly with the help of slot recognition information, but the Slot-Gated model cannot. The examples demonstrate the effectiveness of our Intent-Slot correlation model on intent prediction and slot filling.

Table 3. Results of Ablation Study

Model	ATIS			Snips		
	Slot ($F1$)	Intent (Acc)	Sentence (Acc)	Slot ($F1$)	Intent (Acc)	Sentence (Acc)
Intent-Slot	**95.6**	96.5	86.4	**90.0**	97.4	**78.5**
Slot-Gated (w/o slot recognition)	95.4	95.4	83.7	89.3	96.9	76.4
Intent-Slot (w/o intent attention)	95.4	95.8	84.8	**90.0**	**98.9**	78.1
Intent-Slot (w/o slot attention)	95.5	**97.1**	**86.5**	89.4	96.9	78.2
Intent-Slot (w/o slot gated)	95.5	95.7	84.5	89.6	**98.9**	77.2

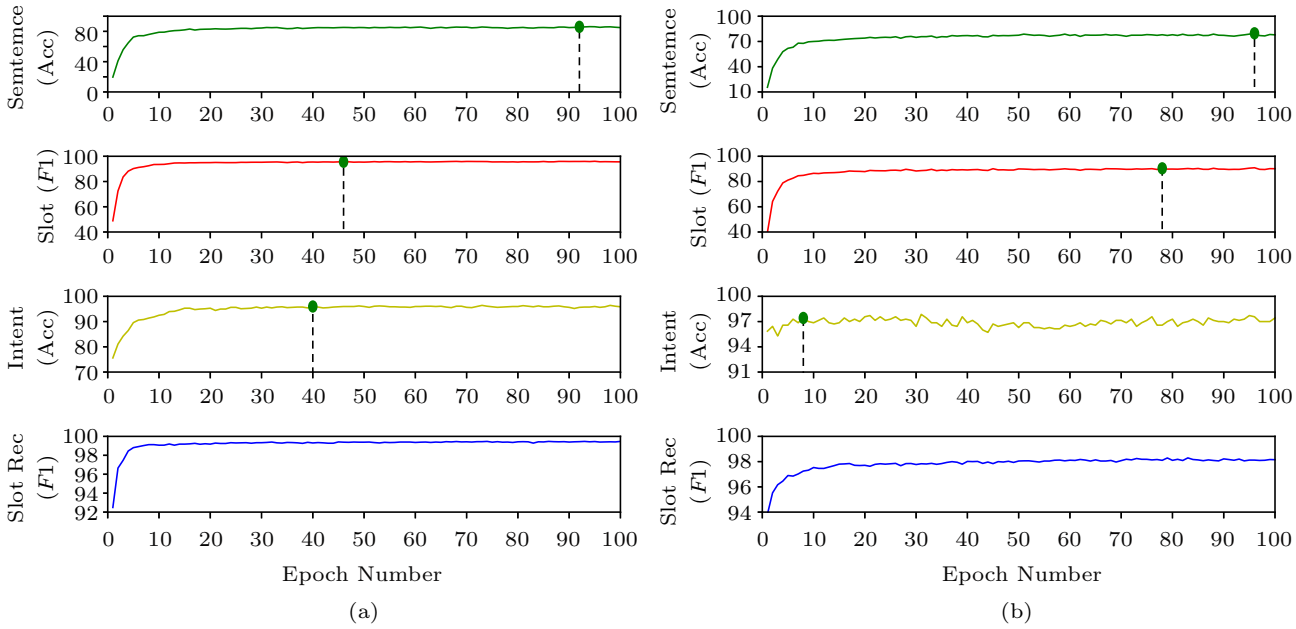

Fig.4. Evaluation results of training epochs on Intent-Slot with the (a) ATIS and (b) Snips datasets. Rec: Recognition.

Case 3 and case 4 show two examples of the ATIS dataset for error prediction of the Intent-Slot model. Case 3 displays that the Intent-Slot model incorrectly recognizes the slot label of "lax" and the intent type of the whole sentence. Case 4 shows that the Intent-Slot model incorrectly recognizes the slot label of "snacks" and the intent type of the whole sentence. It is probably due to the OOV (Out Of Vocabulary) problem of test datasets. For example, with respect to the ATIS dataset, there are 32 words in the test dataset but out of the training dataset. These OOV words are initialized with random embeddings, which may mislead our model in slot recognition and further mislead slot filling and intent prediction. We will further study these bad cases and improve models in the future work.

5 Conclusions

In this paper, we explored a new framework for joint intent prediction and slot filling by integrating slot recognition and Slot-Gated mechanism. Experi-

Table 4. Examples for Case Study

Case	Sentence and Slot Filling									Intent
Case 1	Sentence	what	does	the	restriction		ap58		mean	\
	Slot-Gated	O	O	O	O		O		O	–
	Intent-Slot	O	O	O	O		B-restriction_code		O	–
Case 2	Sentence	Flight	numbers	from	Chicago	to	Seattle	on	continental	\
	Slot-Gated	O	O	O	B-from.city	O	B-to.city	O	B-airline	flight_time
	Intent-Slot	O	O	O	B-from.city	O	B-to.city	O	B-airline	flight_no
Case 3	Sentence (part 1)	List		the	airfare	for	american		airlines lax	\
	Ground truth (part 1)	O		O	O	O	B-airline_name		I-airline_name	airfare#flight
	Intent-Slot (part 1)	O		O	O	O	B-airline_name		I-airline_name	airfare
	Sentence (part 2)	flight		19	from	jkf	to			
	Ground truth (part 2)	O		B-fight_number	O	B-fromloc. airport_code	O		B-toloc. airport_code	
	Intent-Slot (part 2)	O		B-fight_number	O	B-fromloc. airport_code	O		B-toloc. city_name	
Case 4	Sentence	Are		snacks	served	on	tower		air	\
	Ground truth	O		B-meal_description	O	O	B-airline_name		I-airline_name	meal
	Intent-Slot	O		O	O	O	B-airline_name		I-airline_name	flight

Note: Red represents incorrect slot labels or intent types. Slots are tagged in the "BIO" (Begin, Inside, Other) format. \ denotes there is no information and – denotes Intent is not the comparison content of this case.

mental results revealed that our Intent-Slot correlation model can achieve state-of-the-art semantic frame performance.

Currently, we obtained slot recognition, intent prediction and slot filling by assigning the same weight to each task in the joint optimization. In the future, we will study how the weights affect results and improve joint optimization. Besides, we will study the bad cases to improve our design of models, and we will explore the combination of implicit and explicit Intent-Slot correlation modeling to improve performance.

References

[1] Tur G, De Mori R. Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. John Wiley & Sons, 2011. DOI: 10.1002/9781119992691.

[2] Haffner P, Tür G, Wright J H. Optimizing SVMs for complex call classification. In *Proc. the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2003, pp.632-635. DOI: 10.1109/ICASSP.2003.1198860.

[3] Hu J, Wang G, Lochovsky F, Sun J T, Chen Z. Understanding user's query intent with Wikipedia. In *Proc. the 18th International Conference on World Wide Web*, April 2009, pp.471-480. DOI: 10.1145/1526709.1526773.

[4] Sarikaya R, Hinton G E, Ramabhadran B. Deep belief nets for natural language call-routing. In *Proc. the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp.5680-5683. DOI: 10.1109/ICASSP.2011.5947649.

[5] Raymond C, Riccardi G. Generative and discriminative algorithms for spoken language understanding. In *Proc. the 8th Annual Conference of the International Speech Communication Association*, August 2007, pp.1605-1608.

[6] Yao K, Peng B, Zhang Y, Yu D, Zweig G, Shi Y. Spoken language understanding using long short-term memory neural networks. In *Proc. the 2014 IEEE Spoken Language Technology Workshop*, Dec. 2014, pp.189-194. DOI: 10.1109/SLT.2014.7078572.

[7] Guo D, Tur G, Yih W T, Zweig G. Joint semantic utterance classification and slot filling with recursive neural networks. In *Proc. the 2014 IEEE Spoken Language Technology Workshop*, Dec. 2014, pp.554-559. DOI: 10.1109/SLT.2014.7078634.

[8] Hakkani-Tür D, Tür G, Celikyilmaz A, Chen Y N, Gao J, Deng L, Wang Y Y. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Proc. the 17th Annual Conference of the International Speech Communication Association*, Sept. 2016, pp.715-719. DOI: 10.21437/Interspeech.2016-402.

[9] Chen Y N, Hakanni-Tür D, Tur G, Celikyilmaz A, Guo J, Deng L. Syntax or semantics? Knowledge-guided joint semantic frame parsing. In *Proc. the 2016 IEEE Spoken Language Technology Workshop*, Dec. 2016, pp.348-355. DOI: 10.1109/SLT.2016.7846288.

[10] Wang Y, Shen Y, Jin H. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2018, pp.309-314. DOI: 10.18653/v1/N18-2050.

[11] Liu B, Lane I. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proc. the 17th Annual Conference of the International Speech Communication Association*, Sept. 2016, pp.685-689. DOI: 10.21437/Interspeech.2016-1352.

[12] Goo C W, Gao G, Hsu Y K, Huo C L, Chen T C, Hsu K W, Chen Y N. Slot-gated modeling for joint slot filling and intent prediction. In *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2018, pp.753-757. DOI: 10.18653/v1/N18-2118.

[13] Liu B, Lane I. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*, Dec. 2015.

[14] Chen Q, Zhuo Z, Wang W. BERT for joint intent classification and slot filling. arXiv:1902.10909, 2019. https://arxiv.org/abs/1902.10909, August 2020.

[15] Li C, Li L, Qi J. A self-attentive model with gate mechanism for spoken language understanding. In *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, October 31-November 4, 2018, pp.3824-3833. DOI: 10.18653/v1/D18-1417.

[16] Zhang C, Li Y, Du N, Fan W, Philip S Y. Joint slot filling and intent detection via capsule neural networks. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, July 28-August 2, 2019, pp.5259-5267. DOI: 10.18653/v1/P19-1519.

[17] E H H, Niu P, Chen Z, Song M. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, July 28-August 2, 2019, pp.5467-5471. DOI: 10.18653/v1/P19-1544.

[18] Schuster M, Paliwal K K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997, 45(11): 2673-2681. DOI: 10.1109/78.650093.

[19] Tur G, Hakkani-Tür D, Heck L. What is left to be understood in ATIS? In *Proc. the 2010 IEEE Spoken Language Technology Workshop*, Dec. 2010, pp.19-24. DOI: 10.1109/SLT.2010.5700816.

[20] Coucke A, Saade A, Ball A *et al.* Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. arXiv:1805.10190, 2018. https://arxiv.org/abs/1805.10190, August 2020.

Jun-Feng Fan received his M.E. degree in mechanical engineering from Xidian University, Xi'an, in 2015. He is now an algorithm engineer in AI Laboratory of KingSoft Corporation, Beijing. His research interests include knowledge graph and question answering.

Mei-Ling Wang is now a senior algorithm engineer in AI Laboratory, KingSoft Corporation, Beijing. She received her Ph.D. degree in computer science from Institute of Software, Chinese Academy of Sciences, Beijing, in 2015. Her main research interests include knowledge graph and question answering.

Chang-Liang Li received his Ph.D. degree in pattern recognition and intelligence systems from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2015. He is currently the principal of AI Laboratory, KingSoft Corporation, Beijing. His research interests include knowledge graph and machine translation.

Zi-Qiang Zhu received his M.E. degree in software engineering from Jilin University, Jilin, in 2017. He is now an algorithm engineer in AI Laboratory of KingSoft Corporation, Beijing. His research interests include knowledge graph and operating system.

Lu Mao obtained her Ph.D. degree in environmental sciences from Peking University, Beijing, in 2019. She is currently an algorithm engineer in AI Laboratory, KingSoft Corporation, Beijing. Her research interests focus on knowledge graph and few-shot information extraction.