Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2074035, 10 pages
https://doi.org/10.1155/2022/2074035

Research Article

A Joint Model of Natural Language Understanding for Human-Computer Conversation in IoT

Rui Sun,[1] **Lu Rao**,[2] **and Xingfa Zhou**[2]

[1]*School of Artificial Intelligence, Leshan Normal University, Leshan, China*
[2]*AI Lab, Sichuan Changhong Electric Appliance Co., Ltd, Chengdu, China*

Correspondence should be addressed to Lu Rao; lu.rao@changhong.com

Received 1 March 2022; Revised 7 July 2022; Accepted 2 August 2022; Published 21 August 2022

Academic Editor: Xingsi Xue

Natural language understanding (NLU) technologies for human-computer conversation is becoming a hot topic in the Internet of Things (IoT). Intent detection and slot filling are two fundamental NLU subtasks. Current approaches to these two subtasks include joint training methods and pipeline methods. Whether treating intent detection and slot filling as two separate tasks or training the two tasks as a joint model utilizing neural networks, most methods fail to build a complete correlation between the intent and slots. Some studies indicate that the intent and slots have a strong relationship because slots often highly depend on intent and also give clues to intent. Thus, recent joint models connect the two subtasks by sharing an intermediate network representation, but we argue that precise label information from one task is more helpful in improving the performance of another task. It is difficult to achieve complete information interaction between intent and slots because the extracted features in existing methods do not contain sufficient label information. Therefore, a novel bidirectional information transfer model is proposed in order to create a sufficient interaction between intent detection and slot filling with type-aware information enhancement. Such a framework collects more explicit label information from the network's top layer and learns discriminative features from labels. According to the experimental results, our model greatly outperforms previous models and achieves the state-of-the-art performance on the two datasets: ATIS and SNIPS.

1. Introduction

The definition of Internet of Things (IoT) is the network where devices or sensors deploy in physical environments using intelligent interfaces to connect and communicate within different user scenarios [1, 2]. Recent studies show that the natural language will become the primary interactive mode between people and devices in IoT. Human-computer conversation technologies can be used to connect people and a variety of objects in the network, which include natural language understanding, knowledge graph, and semantic web. To combine semantic web technologies with IoT adaptively, Xue et al. [3–5] propose some algorithms to match sensor ontologies for the purpose of implementing the semantic interoperability among intelligent sensor applications. Natural language understanding technologies are also widely utilized in IoT.

Natural language understanding (NLU) is an essential component computer conversation system of IoT, which generally includes intent detection and slot filling. Both tasks focus on determining the user's intention and collect critical constituents via annotating the utterance. There is an example from the ATIS dataset shown in Figure 1. The utterance "I need a flight from los angeles to charlotte today" is annotated by slot labels on a word-level, while intent detection gives at least one intent label to the whole sentence.

Intent detection and slot filling are naturally defined as two separate tasks [6]. Intent detection refers to the classification problem with the method of machine learning such as support vector machines (SVMs) [7] and deep learning frameworks [8–11]. In addition, intent detection utilized in IoT is normally based on keyword extraction. Slot filling can be treated as a sequence labeling task. Conditional random fields (CRF) [12, 13], maximum entropy Markov models (MEMMs) [14], and

Sentence	I	need	a	flight	from	los	angeles	to	charlotte	today
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Slot	O	O	O	O	O	B-fromloc	I-fromloc	O	B-toloc	B-depart_time
Intent	Atis_flight									

Figure 1: The illustration of an utterance annotated with slot labels and intent label. Three rows represent the input sequence, the corresponding slot labels, and the intent label of the input sequence, respectively.

recurrent neural networks (RNNs) [15–19] are widely adopted to forecast the labels of slot. Furthermore, ontology-based methods are also leveraged for slot filling in IoT. The pipeline methods can be used to separately perform the two tasks, but such frameworks may cause error propagation.

To solve the problems caused by pipeline manners, extracting the slot information and determining the utterance's intent are performed with joint learning methods. Previously, neural networks are used to share the sentence-level features between the two subtasks [20–23]. In addition, an attention-based RNN method is proposed to provide additional information to slot filling and intent detection [24]. Although such approaches avoid the problem of error propagation, the interaction of the two tasks is not considered. In fact, the identified intent information may provide clues to slot filling and vice versa. For example, if the utterance is recognized as intent "atis_flight," it is more likely for the word "charlotte" to have a slot of "toloc" than other labels such as "personal_name." On the contrary, with the word "charlotte" is identified as "toloc" and "today" is identified as "depart_date," the utterance is more likely to be annotated by an intent label "atis_fight" rather than "atis_distance." Thus, it can be seen that slots and intent are interactive.

Some existing works learn to establish interaction between slots and intent. Goo et al. [25] apply the intent context vector to the LSTM layer via a slot-gated mechanism. However, the intent detection task does not utilize the slot information, and the information interaction is unidirectional. The Capsule Neural Network [26, 27] is used to maintain hierarchical relationships in slots, intentions, and words [28]. Besides, the association between slots and intent is improved by a SF-ID network [29]. Wang et al. [30] also design a Bi-model structure to perform slot filling and intent detection jointly. These methods utilize network's intermediate information to build correlation between slots and intent. However, we argue that the information extracted from above approaches represents the sentence features and it is insufficient to express the label information. And the joint model performance can be improved by the specific label probability distribution. Extracting sufficient features is difficult between the two tasks. Therefore, it has become a challenge in recent research to extract the explicit label information and establish a complete correlation between the intent and slots. Unlike the previous approaches, our model extracts information from network's top layer, which preserves more explicit label information.

In this paper, we propose a bidirectional information transfer model for joint intent detection and slot filling with type-aware information enhancement. This model seeks to respond to the problem that most approaches do not build complete correlation between the intent and slots. Firstly, to learn the discriminative features from the slots and intent labels, we provide a type-aware mechanism. Secondly, to improve the connection between intent and slots, a unique bidirectional information transfer method is devised that takes advantage of label probability distribution from the network's top layer. As a result, more precise information of labels is collected for propagation. Furthermore, a tagging scheme is introduced to diminish the number of slot types for slot filling. The training process is faster and more efficient compared to the "BIO" format annotation.

We compare our method with some published state-of-the-art models on ATIS [31] dataset and SNIPS dataset. Experimental results show the effectiveness of our framework, which outperforms most of current state-of-the-art models. Especially in slot filling and sentence accuracy, our model achieves 1.3% improvement on SNIPS dataset and 1.2% absolute gain on ATIS dataset. In addition, we believe that we are the first to use label probability distribution collected from network's top layer for predicting slot and intent jointly.

The remainder of this paper is arranged as follows. Firstly, this paper begins by defining the problem statement for our framework in Section 2. Afterwards, our proposed framework is presented in Section 3. Then we introduce the experimental design, present the findings, and analyze our framework in Section 4. The related work is stated in Section 5. Finally, we conclude our work in Section 6 and discuss the direction of future work in Section 7.

2. Problem Statement

We consider the intent detection as an utterance-level classification task, while slot filling refers to the token-level classification task. We forecast the intent label y^{intent} and a sequence of slot labels $(y_i^{\text{start}}, y_i^{\text{end}})$ as outputs from the input sentence $\{x_1, x_2, \cdots, x_T\}$ with T tokens. Especially for slot filling,we design a simple tagging scheme to reduce the number of slot categories, which will be introduced in detail in the next section. The objective is to reduce the discrepancy between the ground-truth data and estimated outputs.

3. Approach

In this section, our bidirectional information transfer framework will be presented in detail. Figure 2 gives an overview

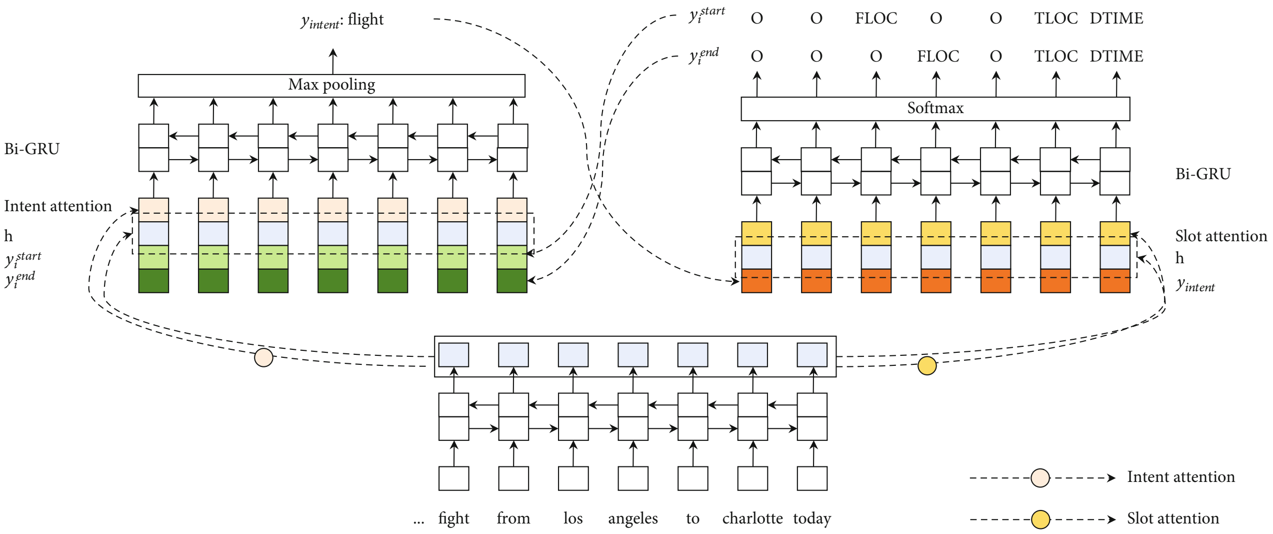


Figure 2: Framework of our bidirectional information transfer method. The framework is comprised of three main parts: the shared encoder, the decoder slot filling, and the intent detection. The type-aware method works on the bottom layer of the two decoders, respectively. The bidirectional information transmission module is utilized between the two decoders, which propagates tag information from top layers of the two decoders. For example, the probability distribution of intent label y_{intent} is transmitted to the bottom layer of slot filling decoder. And the probability distribution of slot labels $[y_i^{\text{start}}, y_i^{\text{end}}]$ is transmitted to the bottom layer of intent detection decoder simultaneously.

of our approach. We can see that the intent detection is transformed into a classification problem, while the slot filling is transformed into two sequence labeling problems. Following that, the tagging scheme is introduced firstly. Then we discuss the input representation for the out-of-vocabulary (OOV) problem and how to enhance the input semantics representation. In addition, a graph recurrent unit (GRU) network is simply introduced, which is utilized as the main layer in our framework. A type-aware information enhancement is used to improve slot filling and intent detection in detail. Lastly, the bidirectional information transfer scheme is presented, and a joint training method is used to optimize both tasks simultaneously.

3.1. Tagging Scheme. Let us consider the slot filling task's tagging scheme first. The slot filling is transformed into two sequence labeling [32] tasks, where the inspiration comes from the Pointer Network [33]. We simplify the tasks as Figure 3 shown.

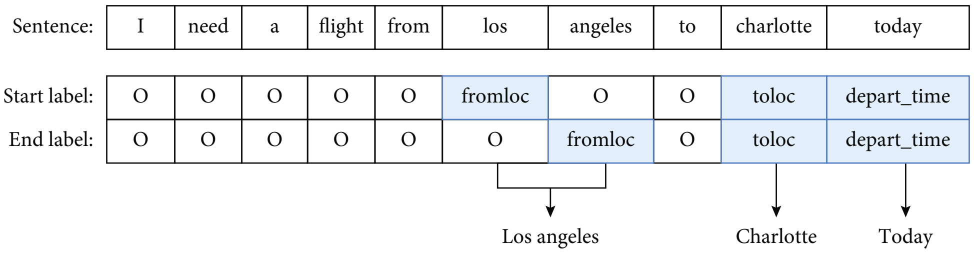
Finding the start position of the entity in an utterance is the primary objective of first task. If a token is the first word in an entity span, it will be annotated with associated slot type. If the token is not the first word, assigning the token with the label "O"(Outside) means it has no significance for the "start sequence labeling." On the contrary, the second task seeks to determine the end position of the entity in an utterance. The labeling process of "end sequence labeling" is similar to the "start sequence labeling," where the difference is to find the end position of an entity span in an utterance.

Figure 3 gives a sample of the tagging method. The words "los," "angeles," "charlotte," and "today" are tagged with the corresponding slot type label using our tagging scheme. Our tagging method is obviously distinct from the "BIO" tagging format. Because we only estimate an entity span's beginning and end position, there are fewer number of slot types. This indicates that our framework is more effective and time-saving. The training process using our tagging method will be discussed in detail in the subsection Slot Filling with Type-aware Information Enhancement.

3.2. Input Representation Layer. A common way to incorporate context information of words is to use input representations learned from unannotated corpora. For most previous studies, word embedding is utilized as a direct input for most language tasks, but it is unable to address the out-of-vocabulary (OOV) problem. As characters are shared across words, the input representation layer maps input sentences into vectors via concatenating word-level embedding and character-level embedding. In this way, unknown words can be generated using their component characters. In addition, we utilize a layer of CNN and a MaxPooling layer to incorporate word character sequence embedding into a dense vector.

The input sequence $\{x_1, x_2, \cdots, x_T\}$ represents the i_{th} word of a sentence with T words. The character-level embedding is represented by e_i^c, and the word-level embedding is represented by e_i^w. Thus, the expression of input representation is $e_i = [e_i^w, e_i^c], e_i \in R^{dc+dw}$, where dc is the character-level embedding dimension and dw is the word-level embedding dimension.

3.3. Bidirectional Graph Recurrent Unit (GRU). The graph recurrent unit (GRU) network is leveraged in the encoder and decoder to extract contextual information and semantic features of an utterance. The GRU network is first proposed by [34] to consider sequence labeling tasks, which has a simpler framework than long-short temporary memory (LSTM) network. The formulation is written as follows [34]:



Sentence:	I	need	a	flight	from	los	angeles	to	charlotte	today
Start label:	O	O	O	O	O	fromloc	O	O	toloc	depart_time
End label:	O	O	O	O	O	O	fromloc	O	toloc	depart_time

Figure 3: A case of tagging method utilized in our framework. Three rows are the input sequence, the star label of an entity span, and the end label of an entity span. In addition, if a word is neither the start word of an entity span nor the end word of an entity span, it will be annotated with the label "O"(Outside).

$$z_i = \sigma(W_z \bullet [h_{i-1}, x_i]), \tag{1}$$

$$r_i = \sigma(W_r \bullet [h_{i-1}, x_i]), \tag{2}$$

$$\widetilde{h_i} = \tanh\left(W \bullet [r_i * h_{i-1}, x_i]\right), \tag{3}$$

$$h_i = (1 - z_i) * h_{i-1} + z_i * \widetilde{h_i}, \tag{4}$$

where σ denotes the sigmoid function, r_i is the reset gate, and z_i represents the update gate. The reset gate decides how to integrate incoming input with the prior memory, while the update date is used to specify how much past memory is preserved to the current time step. Such two gates allow information in long-term sequences to be preserved and not to be cleared over time.

3.4. Intent Detection with Type-Aware Information Enhancement. Normally the detection of intent is viewed as a classification task. Recent methods begin to use deep learning frameworks to accomplish this task [8–11]. Some of them utilize the attention mechanism [35] to focus on partial features. Type information, according to our research, is useful in modeling the learning of discriminative features. So in this paper, to effectively utilize the type information, a straightforward but efficient method named type-aware attention mechanism is suggested. Afterwards, we will detail this mechanism in the intent detection task. As shown in Equation (2), W_{intent} denotes the intent type, and α_i represents the weight of intent attention. For each token, we obtain the hidden state h_i of type-aware intent, which is shown as below:

$$\alpha_i = \text{softmax}\left(e_i U W_{\text{intent}}^T\right), \tag{5}$$

$$h_i^{\text{context}} = \alpha_i W_{\text{intent}}, \tag{6}$$

$$h_i = \left[h_i^{\text{context}}, e_i\right], \tag{7}$$

where $W^T \in R^{N \times d}$ denotes the trainable weight matrix, d_i represents the information vector dimension of intent category, and N denotes the quantity of intent categories. U represents the trainable matrix parameter.

A bidirectional GRU layer is applied to h_i to integrate the utterance representation and intent category information. The input is mapped to each intent category using a liner layer, as shown in Equations (8)–(10). By sharing the W_{intent} matrix, we create a link between the stage of intent detection and the initial layer in Equations (5) and (10).

$$h_i^{\text{intent}} = BiGRU(h_i), \tag{8}$$

$$h^{\text{intent}} = \text{MaxPooling}\left(\left[h_0^{\text{intent}}, \cdots, h_n^{\text{intent}}\right]\right), \tag{9}$$

where h^{intent} represents intent hidden state.

The intent detection is considered as a single-label classification task; thus, the softmax function is used to compute the probability distribution y_{intent} of the intent label:

$$y_{\text{intent}} = \text{softmax}\left(h^{\text{intent}} W_{\text{intent}} + b_{\text{intent}}\right), \tag{10}$$

where W_{intent} and b_{intent} are the trainable matrix parameters.

3.5. Slot Filling with Type-Aware Information Enhancement. Slot filling, like intent detection, also utilizes our type-aware method to extract distinguishing features from slot category. In the process of slot filling, the parameter of start slot category is represented by $W_{\text{slot}}^{\text{start}}$, while the parameter of end slot category is denoted by $W_{\text{slot}}^{\text{end}}$. Because of our unique tagging scheme, the outputs of slot filling is actually divided into two classifiers. Equivalent to Equations (5)–(7), we use a slot type-aware component to calculate s_i. Formally, the tag of i_{th} word w_i when labeling the start position is formulated as Equation (13).

$$s_i^{\text{slot}} = BiGRU(S_i), \tag{11}$$

$$y_i^{\text{start}} = \text{softmax}\left(s_i^{\text{slot}} W_{\text{slot}}^{\text{start}} + b_{\text{slot}}^{\text{start}}\right), \tag{12}$$

$$\text{start\_tag}(w_i) = \text{argmax}_k\left(y_i^{\text{start}} = k\right), \tag{13}$$

where y_i^{start} denotes the slot results of i_{th} word when labeling the start position and $W_{\text{slot}}^{\text{start}}$ and $b_{\text{slot}}^{\text{start}}$ present the trainable matrix parameters.

Analogously, Equation (15) is formulated to compute the entity span's end tag.

$$y_i^{\text{end}} = \text{softmax}\left(s_i^{\text{slot}} W_{\text{slot}}^{\text{end}} + b_{\text{slot}}^{\text{end}}\right), \tag{14}$$

$$\text{end\_tag}(w_i) = \text{argmax}_k\left(y_i^{\text{end}} = k\right), \tag{15}$$

where y_i^{end} denotes the slot results of i_{th} word when labeling the end position and $W_{\text{slot}}^{\text{end}}$ and $b_{\text{slot}}^{\text{end}}$ are the trainable matrix parameters.

3.6. Bidirectional Information Transfer Scheme. Studies have shown that slots and intent are related and can reinforce each other [25, 28, 29]. Recently, intermediate information of the network is used to establish the relationship between slot filling and intent detection. However, the extracted information is insufficient to express the label information. We argue that precise label information from one subtask is more useful to another one. As a result, a two-way information transfer framework is introduced for integrating the two tasks.

During the slot filling process, the probability distribution of intent label y_{intent} is combined with the slot representation s_i . The formulation below is used to replace Equation (11) and showed in Equation (16). Note that the extracted features utilized in our model contain more precise label information, which is collected from network's top layer. After replacement, the slot representation s_i^{slot} will include both intent category information and semantic information of slot.

$$s_i^{\text{slot}} = BiGRU([s_i, y_{\text{intent}}]). \tag{16}$$

Analogously, For the intent detection task, we build a slot to intent iteration component shown as Equations (17)–(19).

$$h_i^{\text{intent}} = BiGRU\left(\left[h_i, y_i^{\text{start}}, y_i^{\text{end}}\right]\right), \tag{17}$$

$$h^{\text{intent\_}} = \text{MaxPooling}\left(\left[h_0^{\text{intent\_}}, \cdots, h_n^{\text{intent\_}}\right]\right), \tag{18}$$

$$y_{\text{intent\_}} = \text{sotfmax}\left(h^{\text{intent\_}} W_{\text{intent\_}} + b_{\text{intent\_}}\right), \tag{19}$$

where $h^{\text{intent\_}}$ is the intent hidden state at each step that contains slot type information y_i^{start} and y_i^{end}. A MaxPooling layer is adopted to reduce the hidden state dimension. Then the probability distribution of intent $y_{\text{intent\_}}$ is calculated using the softmax function.

3.7. Joint Training. A joint training method is adopted to simultaneously update the parameters and learn intent detection and slot filling. The cross-entropy loss for intent detection and slot filling is computed as

$$L_{\text{intent}} = -\sum_{m=1}^{M}\left(\log\left(y_{\text{intent}} = y_{\widehat{\text{intent}}}\right)\right), \tag{20}$$

$$L_{\text{slot}} = -\frac{1}{T}\sum_{i=1}^{T}\sum_{c=1}^{C}\left(\log\left(y_i^{\text{start}} = y_i^{\widehat{\text{start}}}\right) + \log\left(y_i^{\text{end}} = y_i^{\widehat{\text{end}}}\right)\right), \tag{21}$$

where C is the number of slot tags, M represents the number of intent tags, T is the number of words in a sentence. $y_{\widehat{\text{intent}}}$, $\widehat{y_i^{\text{start}}}$, and $\widehat{y_i^{\text{end}}}$ are utilized to denote the ground-truth label of intent and slots.

The training objective is to calculate the minimization of the loss function. Finally, the formulation of the loss function is defined as follows:

$$L = \gamma L_{\text{intent}} + (1-\gamma)L_{\text{slot}}, \tag{22}$$

where γ is applied to adjust the importance of the two tasks.

4. Experiments and Analysis

In this part, we will first describe the experimental datasets, which is shown in Table 1. Then we list several baselines which are compared with our model and describe the training details. Lastly, results and analysis of the proposed bidirectional information transfer model will be stated. In addition, the ablation study is also provided to support the effectiveness of our scheme.

4.1. Dataset. Experiments are conducted using the SNIPS dataset and the ATIS dataset [31]. Both datasets have the annotation of intent and slot labels. The statistics of ATIS and SNIPS datasets are shown in Table 1.

The ATIS dataset, which includes the recordings of people booking flights, is commonly utilized in NLU. There are 500 validation set, 893 test set, and 4478 training set in the dataset. We also make use of SNIPS dataset, which is taken from SNIPS's digital voice assistant, to assess the effectiveness of our algorithms. The SNIPS dataset contains more evenly distributed samples for intent categorization. This dataset is composed of 700 validation set, 700 test set, and 13,084 training set. We divide the dataset in the same way of [28] did in their experiments.

It should be noted that the "BIO" format annotates the original datasets with the header tags "B" and "I," which is removed in our tagging method. Thus, the slot type of SNIPS dataset is modified from 72 to 40, and the slot type of ATIS dataset is modified from 120 to 84 in the actual experiments.

4.2. Baselines. The validity of our framework is verified by comparing it with the latest published models. The following is the list of models:

(i) Joint Seq. [22]: this is a method based on a RNN-LSTM framework to learn intent and slots simultaneously

(ii) Attention-based RNN [24]: the model utilizes an attention-based RNN network to obtain utterance context features for forecasting the intent and slots jointly

(iii) Slot-gated Full Atten [25]: the intent information is applied into the slot filling task by leveraging a slot-gated mechanism

(iv) Capsule-NLU [28]: the Capsule Neural Network [27] is leveraged to connect intent, words, and slots

(v) SF-ID, SF-First [29]: this framework develops a SF-ID network to build correlation between the slots and intent

Table 1: Dataset statistics.

	ATIS	SNIPS
Vocabulary size	722	11241
Slots	120	72
Intents	21	7
Training set size	4478	13084
Development set size	500	700
Testing set size	893	700

(vi) Bi-Model [30]: to implement the interaction between slots and intent, Wang et al. [30] introduce a RNN-based model via semantic parsing framework

(vii) Stack-Propagation [36]: this model develops a joint framework to integrate the intent information with slot filling

Recent works indicate that BERT-based [37] models have also been used successfully to complete the joint tasks [38]. Our method focuses on how to establish interaction between the two tasks instead of the usage of pretrained language model. Thus, our model is not compared with BERT-based models for the sake of fairness.

4.3. Training Details. Word embedding is used as a direct input for most language tasks in earlier studies; however, it is unable to solve the out-of-vocabulary (OOV) issue. To address above problem, the embedding layer in our experiments combines the character-level representation and word-level representation. FastText [39] is used to train the word-level embeddings, while the character-level embeddings are generated by random initialization. In the process of training, we fine-tune the above embeddings. To match the dimension of character-level vectors and word-level vectors, we set up the GRU units as 450. The Adam [40] algorithm is leveraged to optimize the loss function: cross entropy, with a batch size of 64. To decrease overfitting, a dropout of 0.15 is applied to the Bi-GRU. If the accuracy of sentence stops growing after 6 consecutive iterations, we will terminate the training process.

4.4. Results and Analysis

4.4.1. Evaluation Method. We use the F1 score and accuracy in comparison to some published models to assess the effectiveness of our model. Following previous works, the Recall and Precision are utilized to calculate the F1 score. We score a slot as correct if both the entity type and the entity span are correct. An utterance is considered as correct if both the slots and intent are correct. Table 2 shows the main results of our experiments. The first column lists the name of some published models, while the first line shows the datasets utilized in the experiments. Slot (F1) means the slot filling F1 score, the Intent (Acc) means the accuracy of intent detection, and the Overall (Acc) means the accuracy of an utterance.

4.4.2. Main Results. In Table 2, our method is optimal in intent detection and slot filling, achieving an excellent performance on the two datasets: SNIPS and ATIS. On ATIS dataset, our model significantly improves in all 3 aspects comparing with the best model SF-ID, SF-First [29]. The model achieves an absolute gain of 1.2% in terms of sentence accuracy. The experimental results on the SNIPS dataset are also competitive. Our model improves 0.3% in intent detection and 1.3% in slot filling, with better performance than Stack-Propagation [36].

It should be noted that the improved performance of our framework in slot filling and intent detection is mostly due to the efficiency of the proposed bidirectional information transfer method. The findings support the premise that precise label information from one subtask is more helpful to another. As mentioned above, current joint models build the correlation between intent and slots through propagating the information from the intermediate network. However, we argue that more specific label information is contained in the network's top layer. According to the results, we can also find that both datasets have excellent performance in terms of sentence accuracy. This might be because the relationship of slots and intent improves the sentence-level semantic comprehension and enhance the joint model integrality.

Experiments using the pretrained model BERT [37] achieves competitive results compared with current BERT-based models [38]. Our primary research objective in the future will focus on the fusion of BERT and our framework.

4.5. Ablation Study. We perform an ablation experiment to investigate the effects of each module in our framework. The focuses are the two modules: information transfer scheme and type-aware attention mechanism. As Tables 3 and 4 shows, the findings of our entire framework are listed in the second line, while 3 more tests are conducted from line 3 to line 5. The first experiment removes the information transfer scheme and ignores the relationship between intent and slots. The second experiment removes the type-aware method and maintains the information transfer scheme. The third experiment removes both of the two modules mentioned above. Thus, only the pointer-based annotation method is applied.

According to the ablation test of the SNIPS dataset, the slot filling achieves 0.75% improvement in F1 score with the information transfer module (refer to the third row of Table 3. As shown in the third column of Table 3, the intent detection accuracy also increases from 97.85% to 98.29%. The aforementioned results validate the effectiveness of our information transfer module. In addition, the intent detection accuracy achieves a gain of 0.29% and the slot filling F1 score achieves a gain of 0.89% with our type-aware method. Therefore, the type-aware module performs well on both intent detection and slot filling.

In the ablation test of the ATIS dataset, the slot filling F1 score achieves a 1.89% improvement, and the intent detection accuracy increases from 94.84% to 96.97% when utilizing the information transfer module (refer to the forth row of Table 4). In addition, the F1 score also achieves a 2.05% improvement in slot filling, while the intent detection accuracy achieves a 1.57% improvement when utilizing our type-

Table 2: Comparison with published results of joint models on the ATIS and SNIPS datasets.

Model	Overall (Acc)	Intent (Acc)	Slot (F1)	Overall (Acc)	Intent (Acc)	Slot (F1)
Joint Seq. [22]	73.2	96.9	87.3	80.7	92.6	94.3
Attention-based RNN [24]	74.1	96.7	87.8	78.9	91.1	94.2
Slot-gated [25]	75.5	97.0	88.8	82.2	93.6	94.8
Capsule-NLU [28]	80.9	97.3	91.8	83.4	95.0	95.2
SF-ID, SF-First [29]	80.6	97.4	91.4	86.8	97.8	95.8
Bi-Model [30]	83.8	97.2	93.5	85.7	96.4	95.5
Stack-Propagation [36]	86.9	98.0	94.2	86.5	96.9	95.9
Bi-transfer (our model)	85.9	98.3	95.5	88.0	98.7	96.0

Table 3: The ablation test on SNIPS dataset.

	Overall (ACC)	Intent (Acc)	Slot (F1)
Bi-transfer(our model)	85.86	98.29	95.50
Bi-transfer (no information transfer)	83.86	97.85	94.75
Bi-transfer (no type-aware attention)	82.85	98.00	94.61
Bi-transfer (no both component)	81.71	97.57	94.25

Table 4: The ablation test on ATIS dataset.

	Overall (Acc)	Intent (Acc)	Slot (F1)
Bi-transfer(our model)	88.02	98.66	96.00
Bi-transfer (no information transfer)	87.63	96.41	95.88
Bi-transfer (no type-aware attention)	87.12	96.97	95.72
Bi-transfer (no both component)	84.99	94.84	93.83

aware module (refer to the third row of Table 4). The aforementioned experimental results validate the efficiency of the information transfer module and type-aware module. Lastly, we find that the improvements within the ATIS dataset are more absolute than in the SNIPS dataset. This may be because our method is more beneficial to the dataset having a large variety of categories.

Furthermore, we find that the test only utilizing the type-aware module has better performance than the test only utilizing the information transfer module in slot filling. On the contrary, the test only using the information transfer module performs better than the test only using the type-aware module in intent detection. To sum up, our proposed type-aware module works better for slot filling task, which may credit to that this module is beneficial to learning type information.

4.6. Error Analysis. We further analyze some error cases of the experimental results. It is observed that most misclassification cases from the SNIPS dataset include multiple intents, while our framework is designed to connect the slots with single-label intent. In our proposed bidirectional transfer scheme, each token in slot filling is provided with the same multiple intents information, which may import irrelevant noise for some slots. This may influence the global integrality of the joint model, which leads to worse results than the current published models in the sentence accuracy of the SNIPS dataset. In practice, the intent detection module is supposed to provide fine-grained intent information to the token-level slots so that the slot can be guided by its associated intent information. We will concentrate on how to leverage multiple intents to guide corresponding slot predictions in the future.

5. Related Work

The Internet of Things uses devices and sensors to connect various objects to a network, which establishes the correlation between people and the world [1, 2]. Some studies suggest that natural language will become the primary interactive mode between people and terminals in IoT. Human-computer conversation technologies such as natural language understanding, knowledge graph, and semantic web can be used to link people and the physical environments. Xue et al. [3–5] introduce some algorithms to match sensor ontologies for the purpose of applying semantic interoperability among intelligent sensor applications. In addition, natural language understanding technologies are also widely utilized in IoT.

In previous methods, slot filling and intent detection are treated as independent operations in pipeline manners. The task of intent detection is generally regarded as a classification problem, which relies on the methods of support vector machines (SVMs) [7] and deep learning frameworks [8–11]. Recently, a transformer model and universal sentence encoder-based deep averaging network are utilized in intent detection task [9]. Different from intent detection, slot filling is formulated as a task of sequence labeling. Previous work on slot filling is relied on CRF [12] and MEMMs [14]. Currently, neural network models are combined with CRF to address the slot filling issues. Gong et al. [19] perform slot filling task by a deep cascade multitask learning scheme based on BiLSTM-CRF. It is simple to conduct these two tasks separately, but it is difficult to establish the relationship between the slots and intent. Besides, pipeline approaches may result in error propagation issue.

The error propagation issue of pipeline approaches is solved by training the intent detection task and slot filling task jointly [25]. Previous methods share the input embeddings and use a common loss function for joint models [21, 23]. Xu and Sarikaya [20] introduce a CNN-based framework to collect features from slot filling task, which is utilized to enhance the intent detection. A RNN-LSTM architecture [22] is introduced to enable the prediction of intent and slots optimized in a joint model based on bidirectional RNN with LSTM cells. In addition, Liu and Lane [24] develop a RNN-based framework using an attention mechanism to predict the intent and slots jointly. But the approaches mentioned above continue to ignore the correlation of intent and slots.

To address the aforementioned limitation, Goo et al. [25] suggest to use a slot-gated algorithm to connect intent and slots. This method creates an attention scheme to leverage intent information in slot filling. In addition, a stack-propagation model [36] is proposed to merge intent information with the prediction of slots. However, the flow of information in both methods is unidirectional. The Capsule Neural Network [27] is utilized to create correlation between intent, words, and slots [28]. The model adopts a routing-by-agreement mechanism to achieve information transmission. A SF-ID network [29] is suggested to establish bidirectional interaction between intent and slots. Hui et al. [41] also propose a continuous learning model for considering semantic information with various features. Although the methods mentioned above build bidirectional relationship between intent and slots, the features in the process of propagation is still collected from the input representation.

We believe that precise type information can promote both subtasks in joint models. Thus, we develop a two-way information transfer framework with type-aware information enhancement and pointer-based tagging method. We first propose a unique type-aware module to reinforce the discriminative information. Then, we introduce a bidirectional information transmission module to establish complete correlation between slots and intent, which collects precise type information from network's top layer. To accelerate the process of training, a point-based tagging method is leveraged in our model.

6. Conclusion

We introduce a joint model for the prediction of intent and slots with type-aware attention scheme. A bidirectional information transmission module and a type-aware attention module are proposed to create complete correlation between intent and slots, which utilizes the information extracted from network's top layer. Then a point-based tagging scheme is introduced to make the model be more time-saving and efficient. The results of experiments confirm the suggestion that building complete relationship between intent and slots is helpful to promote the performance of the subtasks. In summary, the proposed model outperforms other published models on the SNIPS dataset and ATIS dataset.

7. Future Work

Future research will focus on the fusion method of our framework with language model BERT. It is observed that our method shows poor performance on a dataset that contains multiple intents; thus, we will try to establish a more fine-grained correlation between multiple intents and slot filling. Furthermore, future research should investigate how to combine natural language understanding technologies with devices and sensors.

Data Availability

The article contains all datasets utilized to support the study's conclusions.

Disclosure

This research was previously presented at the 17th International Conference on Computational Intelligence and Security [42]. The corresponding author is Lu Rao.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work is supported by the Scientific Research Projects of Leshan Normal University (Grant No. XJR17001, ZZ201822, and LZD005), the Key Projects of Sichuan Provincial Education Department of China (Grant No. 18ZA0239), the National Key R&D Program of China (Grant No. 2017YFA0700800), and the Projects of Sichuan Tourism Development Research Center (Grant No. LY21-21).

References

[1] L. Tan and N. Wang, "Future internet: the Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Chengdu, China, August 2010.

[2] K. O. M. Salih, T. A. Rashid, D. Radovanovic, and N. Bacanin, "A comprehensive survey on the Internet of Things with the industrial marketplace," 2022, https://arxiv.org/abs/2202.03142.

[3] X. Xue and C. Jiang, "Matching sensor ontologies with multi-context similarity measure and parallel compact differential evolution algorithm," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24570–24578, 2021.

[4] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 10 pages, 2021.

[5] X. Xue and J. Chen, "Using compact evolutionary tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[6] T. Gokhan and D. M. Renato, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, Wiley, 2011.

[7] P. Haffner, G. Tur, and J. H. Wright, "Optimizing SVMs for complex call classification," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, pp. 632–635, Hong Kong, China, 2003.

[8] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. S. Yu, "Zero-shot user intent detection via capsule neural networks," 2018, https://arxiv.org/abs/1809.00385.

[9] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Self-attention networks for intent detection," in *Proceedings of Recent Advances in Natural Language Processing*, pp. 1373–1379, Varna, Bulgaria, October 2019.

[10] E. Okur, S. H. Kumar, S. Sahay, A. A. Esme, and L. Nachman, "Natural language interactions in autonomous vehicles: intent detection and slot filling from passenger utterances," 2019, https://arxiv.org/abs/1904.10500.

[11] Y. Tian and P. J. Gorinski, "Improving end-to-end speech-to-intent classification with reptile," 2020, https://arxiv.org/abs/2008.01994.

[12] L. John, M. C. Andrew, and P. Fernando, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of 18th International Conference on Machine Learning*, pp. 282–289, Williamstown, MA, USA, 2001.

[13] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Interspeech 2007*, pp. 1605–1608, Antwerp, Belgium, August 2007.

[14] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proceedings of 17th International Conference on Machine Learning*, Stanford, CA, USA, 2000.

[15] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 189–194, South Lake Tahoe, NV, USA, December 2014.

[16] G. Mesnil, Y. Dauphin, K. Yao et al., "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

[17] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder LSTM for semantic slot filling," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2077–2083, Austin, Texas, 2016.

[18] B. Liu and I. Lane, "Multi-domain adversarial learning for slot filling in spoken language understanding," 2017, https://arxiv.org/abs/1711.11310.

[19] Y. Gong, X. Luo, Y. Zhu et al., "Deep cascade multi-task learning for slot filling in online shopping assistant," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6465–6472, 2019.

[20] P. Xu and R. Sarikaya, "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, Olomouc, Czech Republic, December 2013.

[21] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 554–559, South Lake Tahoe, NV, USA, December 2014.

[22] D. Hakkani-Tür, G. Tur, A. Celikyilmaz et al., "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Proceedings of the 17th Annual Meeting International Speech Communication Association (INTERSPEECH 2016)*, pp. 715–719, San Francisco, CA, USA, 2016.

[23] Y.-N. Chen, D. Hakkani-Tür, G. Tur, J. Gao, and L. Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding," in *Proceedings of the 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*, San Francisco, CA, USA, 2016.

[24] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," 2016, https://arxiv.org/abs/1609.01454.

[25] C.-W. Goo, G. Gao, Y.-K. Hsu et al., "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 753–757, New Orleans, Louisiana, 2018.

[26] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proceedings of the 21st International Conference on Artificial Neural Networks*, pp. 44–51, Espoo, Finland, 2011.

[27] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of Conference and Workshop on Neural Information Processing Systems*, pp. 3856–3866, Long Beach, CA, USA, 2017.

[28] C. Zhang, Y. Li, N. Du, W. Fan, and P. Yu, "Joint slot filling and intent detection via capsule neural networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5259–5267, Florence, Italy, 2019.

[29] E. Haihong, P. Niu, Z. Chen, and M. 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*, pp. 5467–5471, Florence, Italy, 2019.

[30] Y. Wang, Y. Shen, and H. Jin, "A bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 309–314, New Orleans, Louisiana, 2018.

[31] G. Tur, D. Hakkani-Tur, and L. Heck, "What is left to be understood in ATIS?," in *2010 IEEE Spoken Language Technology Workshop*, pp. 19–24, Berkeley, CA, USA, December 2010.

[32] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8357–8366, Seattle, WA, USA, 2020.

[33] V. Oriol, F. Meire, and J. Navdeep, "Pointer networks," in *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pp. 2692–2700, Montreal, Quebec, Canada, 2015.

[34] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[36] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2078–2087, Hong Kong, China, 2019.

[37] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, https://arxiv.org/abs/1810.04805.

[38] Q. Chen, Z. Zhuo, and W. Wang, “BERT For Joint Intent Classification and Slot Filling,” 2019, https://arxiv.org/abs/1902.10909.

[39] M. Tomas, G. Edouard, B. Piotr, P. Christian, and J. Armand, “Advances in pre-training distributed word representations,” in *Proceedings of LREC*, pp. 8357–8366, Miyazaki, Japan, 2018.

[40] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, https://arxiv.org/abs/1412.6980.

[41] Y. Hui, J. Wang, N. Cheng, F. Yu, T. Wu, and J. Xiao, “Joint intent detection and slot filling based on continual learning model,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, June 2021.

[42] R. Sun, L. Rao, and X. Zhou, “Bidirectional information transfer scheme for joint intent detection and slot filling,” in *2021 17th International Conference on Computational Intelligence and Security (CIS)*, Chengdu, China, November 2021.