

# A Scope Sensitive and Result Attentive Model for Multi-Intent Spoken Language Understanding

Lizhi Cheng<sup>1</sup>, Wenmian Yang<sup>2\*</sup>, Weijia Jia<sup>3\*</sup>

<sup>1</sup>Shanghai Jiao Tong University, Shanghai, PR China

<sup>2</sup>Nanyang Technological University, Singapore, Singapore

<sup>3</sup>BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University (Zhuhai), Guangdong Key Lab of AI and Multi-Modal Data Processing, BNU-HKBU United International College, Zhuhai, Guang Dong, PR China  
clz19960630@sjtu.edu.cn, wenmian.yang@ntu.edu.sg, jiawj@bnu.edu.cn

## Abstract

Multi-Intent Spoken Language Understanding (SLU), a novel and more complex scenario of SLU, is attracting increasing attention. Unlike traditional SLU, each intent in this scenario has its specific scope. Semantic information outside the scope even hinders the prediction, which tremendously increases the difficulty of intent detection. More seriously, guiding slot filling with these inaccurate intent labels suffers error propagation problems, resulting in unsatisfied overall performance. To solve these challenges, in this paper, we propose a novel Scope-Sensitive Result Attention Network (SSRAN) based on Transformer, which contains a Scope Recognizer (SR) and a Result Attention Network (RAN). Scope Recognizer assignments scope information to each token, reducing the distraction of out-of-scope tokens. Result Attention Network effectively utilizes the bidirectional interaction between results of slot filling and intent detection, mitigating the error propagation problem. Experiments on two public datasets indicate that our model significantly improves SLU performance (5.4% and 2.1% on Overall accuracy) over the state-of-the-art baseline.

## Introduction

Spoken Language Understanding (SLU) is a core component of task-oriented dialogue systems, generally containing two subtasks, namely Slot Filling (SF) and Intent Detection (ID) (Tur and De Mori 2011). In traditional SLU tasks, SF is a sequence labeling task aiming to fill in the slot for each token; ID is a sentence-level semantic classification task that identifies the intent label for the entire utterance. Recent studies (Gangadharaiyah and Narayanaswamy 2019; Qin et al. 2020) find that users also express more than one intent in an utterance in many scenarios. Thus, a new SLU task, i.e., Multi-Intent SLU, is derived, attracting increasing attention. A simple example of Multi-Intent SLU is shown in Figure 1.

In Multi-Intent SLU, the input utterance is usually composed of multiple sub-utterances, and each sub-utterance has its own intent. The final ID results are the union of the intent of all sub-utterances. Moreover, the intent is usually closely related to the tokens within its sub-utterance and has a weak

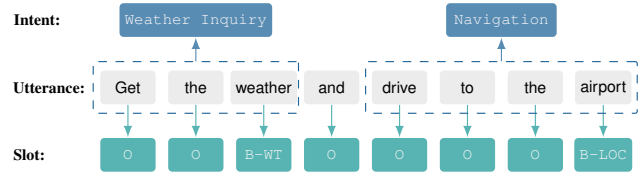


Figure 1: An example of Multi-Intent SLU, where B-WT denotes B-Weather, and B-LOC denotes B-Location.

relationship with the others. For example, as shown in Figure 1, the intent *Weather Inquiry* of the previous sub-utterance is related to the token *Get* and *weather*. And the intent of the second sub-utterance is mainly originated from token *drive* and *airport*, but not *weather*. Thus, in the Multi-Intent SLU task, each intent has its specific scope, and the semantic information outside the scope even negatively affects the ID prediction. Due to the above, the difficulty of ID greatly increased compared with traditional SLU. Therefore, how to divide the scope for each intent effectively to enhance the performance of ID is a major challenge.

Moreover, recent studies (Qin et al. 2019; Cheng, Yang, and Jia 2021; Cheng, Jia, and Yang 2021) find that ID and SF are closely related, and ID results positively promote the predicting performance of SF. Qin et al. (2021, 2020) extend this idea to Multi-Intent SLU and try to utilize the ID results to improve SF via the Graph Attention mechanism. However, with the increase in difficulty, the accuracy of ID in Multi-Intent SLU is not as satisfactory as in traditional SLU. As a result, employing ID results to guide SF will inevitably cause the error propagation problem, affecting SF performance and reducing the overall accuracy of the model. Therefore, how to mitigate the error propagation problem and effectively utilize the correlation between ID and SF to improve the final performance is also a significant challenge.

To solve the above challenges, in this paper, we propose a Scope Sensitive Result Attention Network (SSRAN). Our SSRAN contains a Scope Recognizer (SR) and a Result Attention bi-feedback Network (RAN) based on Transformer. The SR is proposed to capture the scope information for the first challenge. In practice, we first predict preliminary results of ID and SF by the encoder output and obtain

\*Corresponding authors.

their result-embedding vectors. Then we calculate the scope-weighted matrix according to the impact of these results on each token and obtain the scope-sensitive hidden states and result-embedding vectors with this matrix. The SR incorporates scope information into the model and reduces the distraction of out-of-scope tokens.

To further solve the error propagation problem for the second challenge, we design RAN based on the self-attention mechanism. RAN fuses the semantics information of ID and SF results based on their interdependencies (i.e., the semantic similarities of each other) and returns a comprehensive result-semantic vector. We merge the result-semantic vector with the scope-sensitive hidden states to help the following prediction. By RAN, we refine ID and SF bidirectionally, which mitigates the error propagation and overcomes the weakness of one-way refinement (i.e., only ID to SF). Moreover, we design two auxiliary tasks called Intent Number Prediction (INP) and Slot Chunking Task (SCT) for multi-task learning, aiming to improve the performance of ID and SF, respectively.

The main contributions of this paper are presented as follows:

1. We propose SR to enhance the ID accuracy of Multi-Intent SLU, which focuses on the scope of intent and reduces the distraction of out-of-scope tokens.
2. We mitigate the error propagation problem caused by mistaken ID labels to improve overall performance by our RAN, which conducts bidirectional interaction between ID and SF to improve both subtasks.
3. Experimental results on two public datasets show our model is superior to existing SOTA models in terms of all evaluation metrics.

## Related Work

In this section, we introduce the related work from two aspects.

### Slot Filling and Intent Detection

Early studies of SLU (Yao et al. 2014; Mesnil et al. 2014; Peng and Yao 2015; Kurata et al. 2016) prove that predicting SF and ID separately with independent models is less effective than the joint models. Motivated by the above, Goo et al. (2018) firstly propose a gate mechanism to learn the relationship between slot and intent. Inspired by Goo et al. (2018), Haihong et al. (2019); Liu et al. (2019a); Zhang et al. (2019) dive deeper into the relationship between ID and SF and propose some bi-directional networks. The above works mainly focus on the hidden states of the two tasks and lack the result information. Then, Qin et al. (2019) propose a Stack-Propagation framework to utilize the result information, which refines SF with the ID label. Considering the impact of SF results on the ID task, inspired by Yang et al. (2019a), Cheng, Yang, and Jia (2021) propose a bi-feedback network RPFSLU, which guides the second round prediction by the first round results via representing learning. Although RPFSLU performs satisfied, it suffers an extended inference latency caused by predicting SLU results in multiple rounds. To accelerate the inferring process, Wu

et al. (2020) propose a non-autoregressive SlotRefine based on Transformer, which successfully speeds up but encounters the uncoordinated-slot problem. To handle this problem, Cheng, Jia, and Yang (2021) propose LR-Transformer with a Layer Refined Mechanism and a specially designed auxiliary task. Cheng, Jia, and Yang (2022) then expend this work to multi-turn SLU tasks with a Salient History Attention module.

### Multi-Intent SLU

The models above are designed based on the assumption that each utterance only has one single intent. Focusing on the multiple intents scenario, Xu and Sarikaya (2013) and Kim, Ryu, and Lee (2017) begin to study the Multi-Intent SLU. Gangadharaiyah and Narayanaswamy (2019) jointly learning SF and multiple ID via a multi-task framework. Qin et al. (2020) extend their idea in traditional SLU and utilize ID labels in SF with an adaptive interaction network. They upgrade their framework to a non-autoregressive model in (Qin et al. 2021) and achieve the SOTA performance. However, their models are at risk of error propagation due to the increasing difficulty of multiple ID prediction. Compared with their models, we focus on the scope of the intent and reduce the distraction of out-of-scope tokens. Moreover, we also incorporate the impact of SF results on the ID task into the model. These modifications effectively enhance ID accuracy, mitigate error propagation, and improve overall performance.

## Method

### Problem Formulation

In this section, we introduce the problem formulation for the Multi-Intent SLU task.

The input of SLU tasks is an utterance composed by a token sequence  $\mathbf{X} = \{x_1, \dots, x_n\}$ , where  $n$  denotes the sequence length. Given  $\mathbf{X}$  as input, our tasks are composed of Slot Filling (SF) and Intent Detection (ID). Specifically, SF is a sequence labeling task to predict a slot label sequence  $\mathbf{y}^S = \{\mathbf{y}_1^S, \dots, \mathbf{y}_n^S\}$ , while ID is a multi-label semantic classification task to predict the intent labels  $\mathbf{y}^I = \{\mathbf{y}_1^I, \dots, \mathbf{y}_m^I\}$  for the whole utterance, where  $m$  denotes the number of intents in given utterance.

### Overview

In this section, we describe the overview of SSRAN and introduce the relationship between each module.

The general framework of our SSRAN is shown in Figure 2, which consists of four parts: an encoder, a Span Recognizer (SR), a Result Attention Network (RAN), and a decoder. The encoder and decoder are based on the Transformer Encoder Layer (Vaswani et al. 2017) with relative position representations (Shaw, Uszkoreit, and Vaswani 2018); SR and RAN work between the encoder and the decoder, which are the core components of SSRAN.

In practice, we first model the input utterance into a hidden states sequence by our encoder. Then, we get the preliminary SF and ID results according to the hidden states and embed these results. Next, we employ our SR and RAN

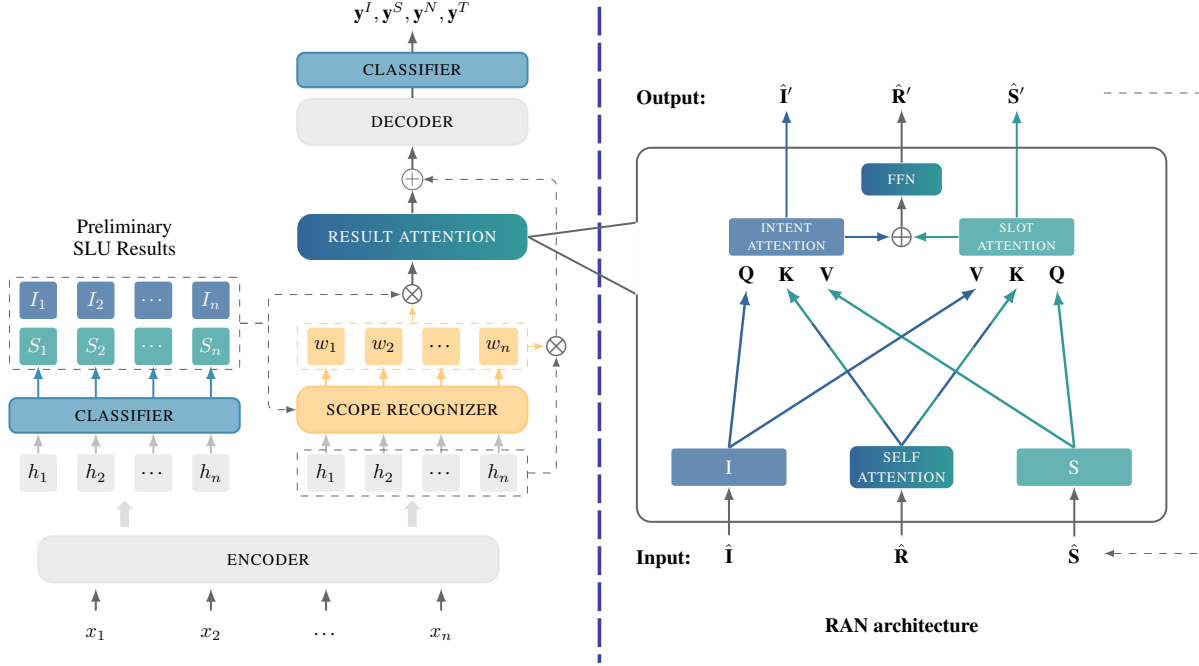


Figure 2: General framework.

to incorporate scope and result information into the model and guide the final prediction via the decoder.

More specifically, in SR, we calculate a scope-weighted matrix, with which we obtain the scope-sensitive hidden states and result embedding vectors. Subsequently, in RAN, we fuse the semantics information of ID and SF results based on their interdependencies via the attention mechanism and return a result-semantic vector. Finally, we merge the result-semantic vector and the scope-sensitive hidden states and obtain the SLU output through our decoder.

In addition, we design two auxiliary tasks, i.e., Intent Number Prediction (INP) and Slot Chunking Task (SCT), for multi-task learning. For INP, we predict the number of intents of each utterance to enhance the ID accuracy. For SCT, we predict the tag of BIO (short for beginning, inside, and outside) of each slot to address sequential dependency.

## Encoder

Recent studies prove SLU results contain specific semantic information, which needs to be considered in prediction. Thus, we employ an encoder to predict preliminary results and incorporate the result semantic information into the model.

Specifically, with the input of  $\mathbf{X} = \{x_1, \dots, x_n\}$ , we first obtain a hidden states sequence  $\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_n \in \mathbb{R}^{d_{model}}\}$ , where  $d_{model}$  is Transformer input and output size. Our encoder is composed of two Transformer Encoder layers. Since the Transformer framework is well known, we do not need to introduce it in every detail.

Subsequently, we predict the preliminary SLU results

$\tilde{\mathbf{y}}^S = \{\tilde{\mathbf{y}}_1^S, \dots, \tilde{\mathbf{y}}_n^S \in \mathbb{R}^{d_s}\}$ ,  $\tilde{\mathbf{y}}^I = \{\tilde{\mathbf{y}}_1^I, \dots, \tilde{\mathbf{y}}_n^I \in \mathbb{R}^{d_i}\}$  by

$$\begin{aligned} \tilde{\mathbf{y}}_j^S &= \mathbf{W}^S (\mathbf{h}_j \oplus \text{Pool}(\mathbf{H})) + \mathbf{b}^S \\ \tilde{\mathbf{y}}_j^I &= \mathbf{W}^I (\mathbf{h}_j \oplus \text{Pool}(\mathbf{H})) + \mathbf{b}^I \end{aligned} \quad (1)$$

where  $\mathbf{W}^S \in \mathbb{R}^{d_s \times 2d_{model}}$  and  $\mathbf{W}^I \in \mathbb{R}^{d_i \times 2d_{model}}$  are fully connected matrices,  $\mathbf{b}^S \in \mathbb{R}^{d_s}$  and  $\mathbf{b}^I \in \mathbb{R}^{d_i}$  are bias vectors,  $d_s$  and  $d_i$  are the categories of the slot labels and intent labels,  $\oplus$  denotes the concatenation operation, and Pool denotes average polling operation.

We then embed these results for the bi-feedback process. Specifically, we obtain SF result embedding vectors  $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_n \in \mathbb{R}^{d_e}\}$  and ID result embedding vectors  $\mathbf{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_n \in \mathbb{R}^{d_e}\}$  by

$$\begin{aligned} \mathbf{S}_j &= \mathbf{E}^S \cdot \text{softmax}(\tilde{\mathbf{y}}_j^S) \\ \mathbf{I}_j &= \mathbf{E}^I \cdot \text{softmax}(\tilde{\mathbf{y}}_j^I) \end{aligned} \quad (2)$$

where  $\mathbf{E}^S \in \mathbb{R}^{d_e \times d_s}$  and  $\mathbf{E}^I \in \mathbb{R}^{d_e \times d_i}$  are fully connected matrices, and  $d_e$  is the embedding size.

## Scope Recognizer

Each intent in Multi-Intent SLU has its specific scope. In this section, we propose the Scope Recognizer (SR) to effectively divide the scope.

Specifically, we first calculate a scope weight matrix  $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ . The scope weight of each token  $\mathbf{w}_j =$

$\{w_{j,1}, \dots, w_{j,n}\}$  is calculated by

$$w_{j,k} = \frac{\exp \alpha_{j,k}}{\sum_{l=1}^n \exp \alpha_{j,l}} \quad (3)$$

$$\alpha_{j,k} = \frac{(\mathbf{h}_j \mathbf{W}^1) ((\mathbf{h}_k + \mathbf{I}_k + \mathbf{S}_k) \mathbf{W}^2)^T}{\sqrt{d_{model}}} \quad (4)$$

where  $\mathbf{W}^1, \mathbf{W}^2 \in \mathbb{R}^{d_{model} \times d_{model}}$  are fully connected matrices.

We then obtain the scope-sensitive hidden states  $\hat{\mathbf{H}} = \{\hat{\mathbf{h}}_1, \dots, \hat{\mathbf{h}}_n \in \mathbb{R}^{d_{model}}\}$  by

$$\hat{\mathbf{h}}_j = \mathbf{h}_j + \sum_{k=1}^n w_k \mathbf{h}_k \quad (5)$$

Similarly, we obtain the scope-sensitive result-embedding vector  $\hat{\mathbf{S}}$  and  $\hat{\mathbf{I}}$  by

$$\hat{\mathbf{S}}_j = \mathbf{S}_j + \sum_{k=1}^n w_k \mathbf{S}_k \quad (6)$$

$$\hat{\mathbf{I}}_j = \mathbf{I}_j + \sum_{k=1}^n w_k \mathbf{I}_k \quad (7)$$

By our SR, we address the scope information for hidden states and result-embedding vectors, which will be further utilized to enhance the final prediction.

### Result Attention Network

As we mentioned in section **Introduction**, considering ID results only in SF cause the error propagation problem and affects the final prediction. To mitigate this problem, in this section, we propose a Result Attention Network(RAN) to effectively utilize the result information of one task in the other.

As shown in Figure 2, RAN is a multi-layer architecture (3 layers in practice) and each layer operates in the same way. The input of the first layer is  $\hat{\mathbf{S}}, \hat{\mathbf{I}}$ , and a initialed result-semantic vector  $\hat{\mathbf{R}} = \hat{\mathbf{I}} + \hat{\mathbf{S}}$ . And each RAN layer returns processed vectors, i.e.,  $\hat{\mathbf{S}}', \hat{\mathbf{I}}'$ , and  $\hat{\mathbf{R}}'$ , which are used as the input of the next RAN layer.

Specifically, we first address  $\hat{\mathbf{R}}$  by

$$\tilde{\mathbf{R}} = \text{Attention}(\hat{\mathbf{R}}, \hat{\mathbf{R}}, \hat{\mathbf{R}}) \quad (8)$$

$$\mathbf{R}^{att} = \text{Norm}(\tilde{\mathbf{R}} + \tilde{\mathbf{R}}) \quad (9)$$

where  $\text{Attention}(q, k, v)$  is the self-attention function and  $\text{Norm}(x)$  is the layer-normalization function used in Transformer (Vaswani et al. 2017).

To mitigate the error propagation problem, we conduct two cross-attention operations between SF and ID to incorporate salient semantic information from both two tasks by

$$\tilde{\mathbf{S}} = \text{Attention}(\hat{\mathbf{S}}, \mathbf{R}^{att}, \hat{\mathbf{I}}) \quad (10)$$

$$\hat{\mathbf{S}}' = \text{Norm}(\hat{\mathbf{S}} + \tilde{\mathbf{S}}) \quad (11)$$

$$\tilde{\mathbf{I}} = \text{Attention}(\hat{\mathbf{I}}, \mathbf{R}^{att}, \hat{\mathbf{S}}) \quad (12)$$

$$\hat{\mathbf{I}}' = \text{Norm}(\hat{\mathbf{I}} + \tilde{\mathbf{I}}) \quad (13)$$

Through this operation, we update the SF and ID result-embedding vectors according to the result semantic information from the opposite task.

Finally, we further utilize these two vectors to update the result-semantic vector  $\tilde{\mathbf{R}}' = \hat{\mathbf{S}}' + \hat{\mathbf{I}}'$  and prevent the vanishing or exploding of gradients by

$$\hat{\mathbf{R}}' = \text{Norm}(\tilde{\mathbf{R}}' + \text{FFN}(\tilde{\mathbf{R}}')) \quad (14)$$

where FFN is a feed-forward network consisting of two linear transformations with a ReLU activation function in between.

RAN finally returns a result-semantic vector  $\mathbf{R} = \{\mathbf{R}_1, \dots, \mathbf{R}_n \in \mathbb{R}^{d_{model}}\}$  from the output of the last RAN layer, which contains both semantic information in SF results and ID results.

### Decoder

With  $\mathbf{R}$  and scope sensitive hidden states  $\hat{\mathbf{H}}$ , we obtain a comprehensive hidden states sequence  $\mathbf{H}^{rs} = \{\mathbf{h}_1^{rs}, \dots, \mathbf{h}_n^{rs} \in \mathbb{R}^{d_{model}}\}$  by

$$\mathbf{H}^{rs} = \text{Norm}(\hat{\mathbf{H}} + \mathbf{R} + \text{FFN}(\text{Pool}(\hat{\mathbf{H}}))) \quad (15)$$

By this operation,  $\mathbf{H}^{rs}$  is both result-semantic-attentive and scope-sensitive.

Then we obtain a hidden states sequence  $\mathbf{H}^d = \{\mathbf{h}_1^d, \dots, \mathbf{h}_n^d \in \mathbb{R}^{d_{model}}\}$  through our decoder, which is composed of four Transformer Encoder layers.

Finally, we predict the SF result and ID result of each token by

$$\begin{aligned} \hat{\mathbf{y}}_j^S &= \mathbf{W}^S (\mathbf{h}_j^d \oplus \text{Pool}(\mathbf{H}^d)) + \mathbf{b}^S \\ \hat{\mathbf{y}}_j^I &= \mathbf{W}^I (\mathbf{h}_j^d \oplus \text{Pool}(\mathbf{H}^d)) + \mathbf{b}^I \end{aligned} \quad (16)$$

Notably, we utilize the same classifier to predict the preliminary result and the final result.

The joint loss function of SLU is defined as:

$$\mathcal{L}_{SF} = - \sum_{j=1}^n \mathbf{y}_j^S \log(\text{softmax}(\hat{\mathbf{y}}_j^S)) \quad (17)$$

$$\mathcal{L}_{ID} = - \sum_{j=1}^n \text{BCE}(\mathbf{y}_j^I, \sigma(\hat{\mathbf{y}}_j^I)) \quad (18)$$

$$\mathcal{L}_{SLU} = \alpha \mathcal{L}_{SF} + (1 - \alpha) \mathcal{L}_{ID} \quad (19)$$

where  $\mathbf{y}_j^S$  and  $\mathbf{y}_j^I$  are the one-hot vector of ground truth labels,  $\alpha$  is a hyper-parameter,  $\sigma$  is the sigmoid function, and BCE is defined as:

$$\text{BCE}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbf{y} \log(\hat{\mathbf{y}}) + (1 - \mathbf{y}) \log(1 - \hat{\mathbf{y}}) \quad (20)$$

During inference, we obtain the label of SF by

$$o_j^S = \text{argmax}(\hat{\mathbf{y}}_j^S) \quad (21)$$

and the label of ID by

$$o^I = \text{Topk} \left( \sum_{j=1}^N \text{softmax}(\hat{\mathbf{y}}_j^I) \right) \quad (22)$$

where Topk indicates to select  $k$  labels with the top probabilities. We determine  $k$  by our INP task that will be introduced in the next section.

## Auxiliary Task

In this section, we design Intent Number Prediction (INP) and Slot Chunking Task (SCT) to further enhance the SLU performance.

Specifically, we predict the INP result  $\hat{\mathbf{y}}^N \in \mathbb{R}^{d_N}$  with our decoder output  $\mathbf{H}^d$  by

$$\hat{\mathbf{y}}^N = \mathbf{W}^N \cdot \text{Pool}(\mathbf{H}^d) + \mathbf{b}^N \quad (23)$$

where  $\mathbf{W}^N \in \mathbb{R}^{d_N \times d_{model}}$  is a fully connected matrix,  $\mathbf{b}^N \in \mathbb{R}^{d_N}$  is a bias vector, and  $d_N$  is equal to  $d_i$ .

And the loss function of INP is defined as:

$$\mathcal{L}_{INP} = -\mathbf{y}^N \log(\text{softmax}(\hat{\mathbf{y}}^N)) \quad (24)$$

where  $\mathbf{y}^N$  is the one-hot vector of the ground truth label of INP.

During inference, we determine the parameter  $k$  in Eq.22 by

$$k = \arg \max(\hat{\mathbf{y}}^N) \quad (25)$$

Besides, previous studies (Wu et al. 2020; Cheng, Jia, and Yang 2021) find that SF suffers the uncoordinated slot problem when using Transformer based models. Since SF utilizes the ‘‘Beginning-Inside-Outside (BIO)’’ tagging format, which clearly divides the slot chunk, we design SCT to attain sequential dependency and enhance SF performance according to the BIO tagging.

In practice, we predict the SCT result  $\hat{\mathbf{y}}^T = \{\hat{\mathbf{y}}_1^T, \dots, \hat{\mathbf{y}}_n^T \in \mathbb{R}^{d_N}\}$  with our decoder output  $\mathbf{H}^d$  by

$$\hat{\mathbf{y}}_j^T = \text{softmax}(\mathbf{W}^T \mathbf{H}_j^d + \mathbf{b}^T) \quad (26)$$

And the loss function of SCT is defined as:

$$\mathcal{L}_{SCT} = - \sum_{j=1}^n \mathbf{y}_j^T \log(\text{softmax}(\hat{\mathbf{y}}_j^T)) \quad (27)$$

where  $\mathbf{y}^T$  is the one-hot vector of the ground truth label, i.e., one of O, B-Tag and I-tag.

We utilize INP and SCT for multi-task learning and the joint loss function of our whole task is

$$\mathcal{L} = \mathcal{L}_{SLU} + \lambda(\mathcal{L}_{INP} + \mathcal{L}_{SCT}) \quad (28)$$

where  $\lambda$  is a hyper-parameter.

## Experiment

In this section, we first introduce the experiment setup. Then, we show the experiment results and conduct ablation studies. Next, we provide a case study and visualization. Finally, we analyze the effect of pre-trained models on SS-RAN.

### Experimental Settings and Baselines

**Dataset:** To evaluate the efficiency of our proposed model, we conduct experiments on two public datasets, i.e., Mix-ATIS and MixSNIPS (Hemphill, Godfrey, and Doddington 1990; Coucke et al. 2018; Qin et al. 2020). The dataset statistics are shown in Table 1.

Dataset	MixATIS	MixSNIPS
Vocabulary Size	722	11241
Intent categories	17	6
Slot categories	116	71
Training set size	13162	39776
Validation set size	756	2198
Test set size	828	2199

Table 1: Dataset statistics.

**Evaluation Metrics:** Following previous work, we evaluate the SLU performance of ID by accuracy and the performance of SF by the F1 score. Besides, we utilize overall accuracy to indicate the proportion of utterances whose slots and intents are both correctly predicted.

**Set up:** Following previous work, we use Adam (Kingma and Ba 2014) to optimize the parameters in our model and adopt the suggested learning rate of 0.001. The batch size is set to 32 according to the training data size. We choose Transformer input and output size  $d_{model}$  as 128, the size of the inner-layer in the feed-forward network  $d_{ff}$  as 512, the number of attention heads as 8, and the dropout ratio as 0.1. The hyper-parameter  $\alpha$  used in Eq.19 is set as 0.65, and  $\lambda$  in Eq.28 is set as 0.3, respectively. When tuning hyper-parameters, we repeat the model 5 times and select the parameters with the best average performance on the validation set as the optimal.

**Baselines:** We compare our model with the existing baselines, including:

- Stack-Propagation (Qin et al. 2019): A LSTM based model with stack-propagation framework and token-level ID.
- Joint Multiple ID-SF (Gangadharaiah and Narayanaswamy 2019): A multi-task framework with the slot-gated mechanism for multiple intent detection and slot filling.
- AG-IF (Qin et al. 2020): A LSTM-based joint model with an adaptive interaction network.
- GL-GIN (Qin et al. 2021): A LSTM-based joint model with a Global-Locally Graph Interaction Network, which is the current SOTA Multi-SLU model.
- LR-Transformer (Cheng, Jia, and Yang 2021): A Transformer based non-autoregressive model with a layer-refined mechanism.
- Basic Model (Vaswani et al. 2017): A six-layers Transformer Encoder, which is equivalent to only utilizing the encoder and decoder of our model.

### Result and Analysis

In this section, we show the results of our experiments and do some analysis.

The experiment results of our model are shown in Table 2. Our model significantly outperforms all the baselines and achieves the best performance in all three metrics. Compared with the SOTA baseline GL-GIN, our model enhances

Model	MixATIS			MixSNIPS		
	Intent	Slot	Overall	Intent	Slot	Overall
Stack-Propagation (Qin et al. 2019)	72.1	87.8	40.1	96.0	94.2	72.9
Joint Multiple ID-SF (Gangadharaiah and Narayanaswamy 2019)	73.4	84.6	36.1	95.1	90.6	62.9
AG-IF (Qin et al. 2020)	74.4	86.7	40.8	95.1	94.2	74.2
GL-GIN (Qin et al. 2021)	76.3	88.3	43.5	95.6	94.9	75.4
LR-Transformer (Cheng, Jia, and Yang 2021)	76.1	88.0	43.3	95.6	94.4	74.9
Basic model	75.1	85.2	41.2	95.2	94.2	73.2
SSRAN w/o SR	76.5	88.7	46.3	96.9	95.4	76.7
SSRAN w/o RAN	76.9	87.6	43.0	96.9	93.9	75.1
SSRAN w/o Auxiliary Tasks	77.1	89.1	47.2	97.6	95.4	77.2
SSRAN	<b>77.9<math>\uparrow</math></b>	<b>89.4<math>\uparrow</math></b>	<b>48.9<math>\uparrow</math></b>	<b>98.4<math>\uparrow</math></b>	<b>95.8<math>\uparrow</math></b>	<b>77.5<math>\uparrow</math></b>

Table 2: SLU performance on MixATIS and MixSNIPS datasets. The numbers with  $\uparrow$  indicate that the improvement of our model over all baselines is statistically significant with  $p < 0.05$  under t-test.

the performance by 1.6%(ID), 1.1%(SF), and 5.4%(Overall) on MixATIS, and 2.8%(ID), 0.9%(SF), and 2.1%(Overall) on MixSNIPS. Compared with the prior Transformer-based model LR-Transformer, we enhance the performance by 1.8%(ID), 1.4%(SF), and 5.6%(Overall) on MixATIS, and 2.8%(ID), 1.4%(SF), and 2.6%(Overall) on MixSNIPS. These results verify the effectiveness of our model intuitively.

Notably, our basic model performs worse than both GL-GIN and LR-Transformer, but SSRAN outperforms both of them, especially on overall accuracy (5.4% on MixATIS). We attribute this enhancement to the fact that SR captures the scope information, bringing more accurate information to each token. And RAN incorporates the result information of both SF and ID into the model, which mitigates the error propagation effectively.

### Ablation Study

In this section, we do an ablation study to verify the effectiveness of SR, RAN and our auxiliary task in detail. The result is shown in Table 2.

**Effect of Scope Recognizer:** We first remove our SR component to verify its effectiveness, which is referred to as SSRAN w/o SR in Table 2. Without SR, the performance drops by 1.4%(ID), 0.7%(SF), and 2.6%(Overall) on MixATIS and 1.5%(ID), 0.4%(SF), and 0.8%(Overall) on MixSNIPS. This phenomenon indicates that scope information influences the accuracy of Multi-Intent SLU, especially on the ID task.

Moreover, as shown in Table 2, although both our model and LR-Transformer (directly adding result-embedding to hidden states) consider the bidirectional interaction, SSRAN performs much better than LR-Transformer. We attribute this to the ability of SR to reduce the distraction of the result information outside the scope.

**Effect of Result Attention Network:** To verify the effect of RAN, we remove our RAN component, which is referred to as texttSSRAN w/o RAN in Table 2. Without RAN, the slot f1 score drops by 1.8% (MixATIS) and 1.4% (MixSNIPS), and the overall accuracy drops by 5.9% and 2.4%, demonstrating the significance of result information.

Note that our model outperforms GL-GIN (only considering ID results in SF) in all metrics, even only utilizing RAN (i.e., SSRAN w/o SR). We attribute this to the fact that RAN considers the bidirectional interaction (also uses SF result in ID), which mitigates the error propagation problem caused by error ID labels.

**Effect of Auxiliary Task:** We finally conduct experiments to find out the effect of the auxiliary task. The results in Table 2 show that the performance of all three metrics drops on both datasets, indicating that our auxiliary tasks further improve our model by multi-task learning. We attribute this improvement to the fact that SCT obtains the sequence dependency and handles the uncoordinated slot problem to help the SF prediction, and the INP task captures ID number information in the utterance level to help the ID prediction.

Model	MixATIS	MixSNIPS
GL-GIN	76.3	95.6
SSRAN (Threshold)	77.1	97.6
SSRAN (TopK-INP)	77.9	98.4
SSRAN (TopK-True)	78.1	98.5

Table 3: ID performance for different predicting methods on MixATIS and MixSNIPS datasets.

**Effect of Topk Method:** Moreover, the SOTA baseline GL-GIN utilizes a manually set threshold to select the predicted intent labels, which is different from ours. Thus, we conduct the experiments to show the performance of our model when utilizing threshold, Topk with INP result, and Topk with the ground-truth intent numbers, to make a more fair comparison. The results are shown in Table 3, where we have the following observations: First, our model still performs better than GL-GIN on ID (0.8% on MixATIS and 2.0% on MixSNIPS) even utilizing threshold, which indicates the effectiveness of our model. Second, compared with selecting ID results by the threshold, utilizing the Topk method further enhances the ID accuracy, which is even close to the ground truth.

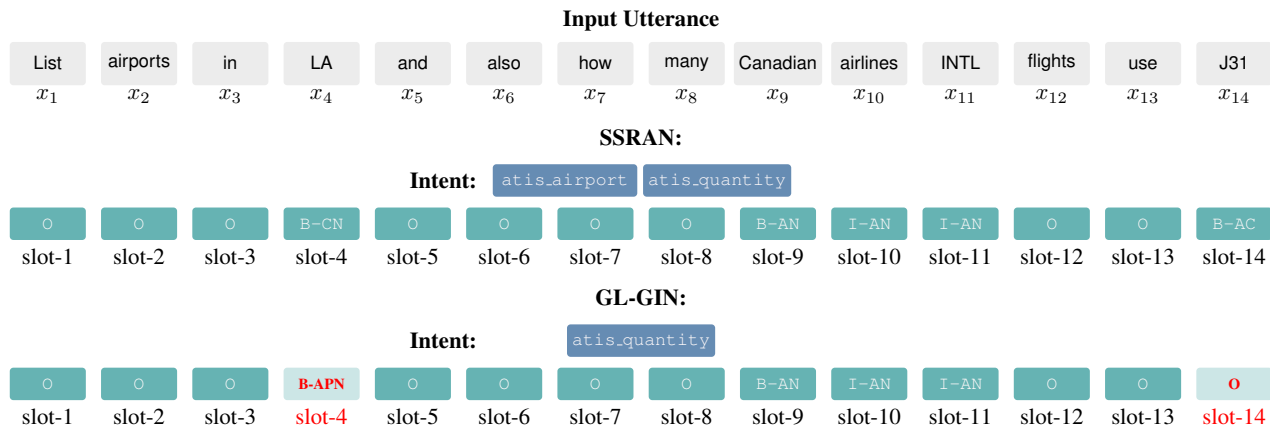


Figure 3: Case study for Multi-Intent SLU, where CN indicates city\_name, AN indicates airline\_name, AC indicates aircraft\_code and APN indicates airport\_name. The slot with red color indicates error-predicted slot label.

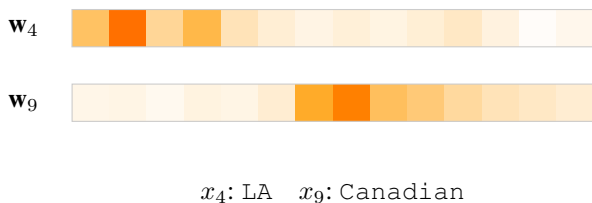


Figure 4: Visualization for scope weight matrix  $W$  in SR. For each column, the darker the color, the bigger the weight.

### Case Study and Visualization

We provide a case study from the MixATIS dataset to compare the SLU results generated by GL-GIN and our model, where the input utterance consists of two sub-utterances separated by the token `and`. As shown in Figure 3, GL-GIN misses the intent `atis.airport` due to lacking scope information and further incorrectly predicts the slot of the token `LA` due to the error propagation. In contrast, SSRAN predicts all the slot labels correctly. We attribute this to the fact that our model captures the scope information by SR and effectively mitigates error propagation by RAN, resulting in a more accurate prediction.

Moreover, we provide visualization for the scope weight matrix computed by SR. As shown in Figure 4, token `LA` will pay more attention to the information from the first sub-utterance, especially from the token `airport`, and predict the intent `atis.airport`. On the contrary, `Canadian` will pay more attention to the second sub-utterance and predict the intent `atis.quantity` with the information from the tokens `how` and `many`.

### Effect of Pretraining

In this section, we conduct experiments to evaluate the ability of our model to combine pretrained models by replacing our encoder with RoBERTa (Liu et al. 2019b), BERT (Devlin et al. 2019), and XLNet (Yang et al. 2019b), respectively.

Model	MixATIS	MixSNIPS
RoBERTa	49.7	80.2
AG-IF + RoBERTa	50.0	80.7
GL-GIN + RoBERTa	53.6	82.6
SSRAN + RoBERTa	54.4	83.1
BERT	51.6	83.0
SSRAN + BERT	54.8	84.5
XLNet	52.1	84.8
SSRAN + XLNet	55.3	85.6

Table 4: Overall performance for pre-trained models on MixATIS and MixSNIPS datasets.

The experiment results are shown in Table 4, where we have two observations. First, our model is benefited from all three pre-train models, which verifies the compatibility between our model and the pre-trained model. Second, SSRAN outperforms AG-IF and GL-GIN when utilizing RoBERTa, which further verifies the effectiveness of our model on Multi-Intent SLU tasks.

### Conclusion

In this paper, we proposed a scope-sensitive model SSRAN for Multi-Intent Spoken Language Understanding task, consisting of a Scope Recognizer and a Result Attention Network. SR focuses on the scope of the intent and guides the prediction by emphasizing closely related information and reducing the distraction of out-of-scope tokens. RAN effectively mitigates the error propagation problem by utilizing the bidirectional interaction between ID and SF. Experimental results show our model achieves the best performance on two public datasets.

### Acknowledgements

This work was supported in part by National Key R&D Program of China (2022YFE0201400); in part by Guangdong Key Lab of AI and Multi-modal Data Processing, United International College (UIC), Zhuhai under Grant No.

2020KSYS007 sponsored by Guangdong Provincial Department of Education; in part by the Chinese National Research Fund (NSFC) under Grant Nos. 62272050, 61872239; in part by Institute of Artificial Intelligence and Future Networks (BNU-Zhuhai) and Engineering Center of AI and Future Education, Guangdong Provincial Department of Science and Technology, China; Zhuhai Science-Tech Innovation Bureau under Grant Nos: ZH22017001210119PWC and 28712217900001, and in part by the Interdisciplinary Intelligence SuperComputer Center of Beijing Normal University (Zhuhai).

## References

- Cheng, L.; Jia, W.; and Yang, W. 2021. An Effective Non-Autoregressive Model for Spoken Language Understanding. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.
- Cheng, L.; Jia, W.; and Yang, W. 2022. Capture Salient Historical Information: A Fast and Accurate Non-Autoregressive Model for Multi-Turn Spoken Language Understanding. *ACM Transactions on Information Systems*.
- Cheng, L.; Yang, W.; and Jia, W. 2021. A Result based Portable Framework for Spoken Language Understanding. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE.
- Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.
- Gangadharaiyah, R.; and Narayanaswamy, B. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 564–569.
- Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. 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)*, 753–757.
- Haihong, E.; Niu, P.; Chen, Z.; and Song, M. 2019. 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*, 5467–5471.
- Hemphill, C. T.; Godfrey, J. J.; and Doddington, G. R. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Kim, B.; Ryu, S.; and Lee, G. G. 2017. Two-stage multi-intent detection for spoken language understanding. *Multi-media Tools and Applications*, 76(9): 11377–11390.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kurata, G.; Xiang, B.; Zhou, B.; and Yu, M. 2016. Leveraging Sentence-level Information with Encoder LSTM for Semantic Slot Filling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2077–2083.
- Liu, Y.; Meng, F.; Zhang, J.; Zhou, J.; Chen, Y.; and Xu, J. 2019a. CM-Net: A Novel Collaborative Memory Network 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)*, 1050–1059.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 530–539.
- Peng, B.; and Yao, K. 2015. Recurrent neural networks with external memory for language understanding. *arXiv preprint arXiv:1506.00195*.
- Qin, L.; Che, W.; Li, Y.; Wen, H.; and Liu, T. 2019. 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)*, 2078–2087.
- Qin, L.; Wei, F.; Xie, T.; Xu, X.; Che, W.; and Liu, T. 2021. GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 178–188.
- Qin, L.; Xu, X.; Che, W.; and Liu, T. 2020. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1807–1816.
- Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 464–468.
- Tur, G.; and De Mori, R. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.



- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Wu, D.; Ding, L.; Lu, F.; and Xie, J. 2020. SlotRefine: A Fast Non-Autoregressive Model for Joint Intent Detection and Slot Filling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1932–1937.
- Xu, P.; and Sarikaya, R. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 78–83. IEEE.
- Yang, W.; Jia, W.; Zhou, X.; and Luo, Y. 2019a. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 4085–4091.
- Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.
- Yao, K.; Peng, B.; Zhang, Y.; Yu, D.; Zweig, G.; and Shi, Y. 2014. Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, 189–194. IEEE.
- Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. S. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5259–5267.