AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling

Libo Qin, Xiao Xu, Wanxiang Che*, Ting Liu
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{lbqin,xxu,car,tliu}@ir.hit.edu.cn

Abstract

In real-world scenarios, users usually have multiple intents in the same utterance. Unfortunately, most spoken language understanding (SLU) models either mainly focused on the single intent scenario, or simply incorporated an overall intent context vector for all tokens, ignoring the fine-grained multiple intents information integration for token-level slot prediction. In this paper, we propose an **A**daptive **G**raph-**I**nteractive **F**ramework (AGIF) for joint multiple intent detection and slot filling, where we introduce an intent-slot graph interaction layer to model the strong correlation between the slot and intents. Such an interaction layer is applied to each token adaptively, which has the advantage to automatically extract the relevant intents information, making a fine-grained intent information integration for the token-level slot prediction. Experimental results on three multi-intent datasets show that our framework obtains substantial improvement and achieves the state-of-the-art performance. In addition, our framework achieves new state-of-the-art performance on two single-intent datasets.

1 Introduction

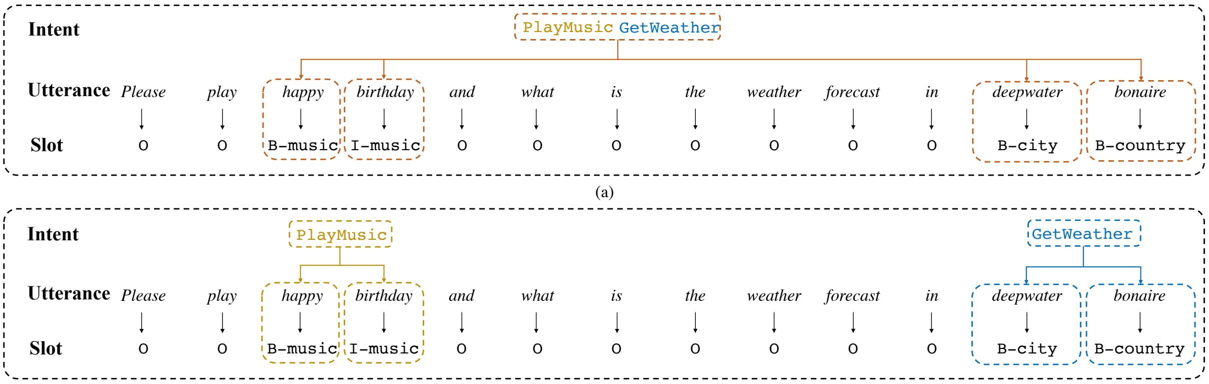
Spoken language understanding (SLU) (Young et al., 2013) is a core component of task-oriented dialog systems. It consists of two typical subtasks, intent detection and slot filling (Tur and De Mori, 2011). Take the utterance "*Please play happy birthday*" for example, the intent detection can be seen as a classification task to classify the intent label (*i.e.*, `PlayMusic`) while the slot filling can be treated as a sequence labeling task to predict the slot label sequence (*i.e.*, `O`, `O`, `B-music`, `I-music`). Dominant SLU systems in the literature (Goo et al., 2018; Li et al., 2018; E et al., 2019; Liu et al., 2019; Qin et al., 2019) adopt joint models to model the relation between the two tasks, which is a direction we follow.

Though achieving promising performances, most prior work only focus on the simple single intent scenario. Their models are trained based on the assumption that each utterance only has one single intent. Actually, users usually express multiple intents in an utterance and Gangadharaiah and Narayanaswamy (2019) shows that 52% of examples are multi-intent in the amazon internal dataset. Nevertheless, the existing trained single intent SLU models fail to effectively handle the multi-intent settings with the original network structure. Ideally, when an SLU system meets an utterance with multiple intents, as shown in Figure 1(a), the model should directly detect its all intents (`PlayMusic` and `GetWeather`). Hence, it is important to consider multi-intent SLU.

Unlike the prior single intent SLU model which can simply leverage the utterance's single intent to guide slot prediction (Goo et al., 2018; Qin et al., 2019), multi-intent SLU faces to multiple intents and presents a unique challenge that is worth studying: how to effectively incorporate multiple intents information to lead the slot prediction. To this end, Gangadharaiah and Narayanaswamy (2019) first explored the multi-task framework with the slot-gated mechanism (Goo et al., 2018) for joint multiple intent detection and slot filling. Their model incorporated intent information by simply treating an intent context vector as multiple intents information. While this is a direct method for incorporating multiple intents information, it does not offer fine-grained intent information integration for token-level slot filling in the sense that each token is guided with the same complex intents information, which is shown in Figure 1(a). In addition, providing the same intent information for all tokens may introduce ambiguity, where it's hard for each token to capture the related intent information. As

[*] Email corresponding.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1807–1816
November 16 - 20, 2020.


Intent
PlayMusic GetWeather
Utterance
Please play happy birthday and what is the weather forecast in deepwater bonaire
Slot
O O B-music I-music O O O O O O O B-city B-country
Intent
PlayMusic
GetWeather
Utterance
Please play happy birthday and what is the weather forecast in deepwater bonaire
Slot
O O B-music I-music O O O O O O O B-city B-country

(a)

(b)

Figure 1: Prior model simply treat multiple intents as an overall intent information (a) vs. our fine-grained multiple intents integration method (b).

shown in Figure 1(b), these tokens "*happy birthday*" should focus on the intent "PlayMusic" while tokens "*deepwater bonaire*" depend on the intent "GetWeather". Thus, each token should focus on the corresponding intent and it's critical to make a fine-grained intent information integration for the token-level slot prediction.

In this paper, we propose an **A**daptive **G**raph-**I**nteractive **F**ramework (AGIF) to address the aforementioned concern. The core module is the proposed adaptive intent-slot graph interaction layer, which is constructed of each token's hidden state of slot filling decoder and embeddings of predicted multiple intents. In this graph, each token's slot node directly connects all predicted intent nodes to explicitly build the correlation between slot and intents. Such an interaction graph is applied to each token adaptively, which make each token has the ability to capture different relevant intent information so that fine-grained multiple intents integration can be achieved. In contrast to prior work simply incorporate multiple intents information statically where the same intents information is used for guiding all tokens, our intent-slot interaction graph is constructed adaptively with graph attention network over each token. This encourages our model to automatically filter the irrelevant information and capture important intent at the token-level.

We first conduct experiments on the multi-intent benchmark dataset DSTC4 (Schuster et al., 2019). Then, to verify the generalization of our framework, we empirically construct two large-scale multi-intent datasets MixATIS (Hemphill et al., 1990) and MixSNIPS (Coucke et al., 2018). The results of these experiments show the effectiveness of our framework by outperforming the current state-of-the-art method. To the best of our knowledge, there are no public large-scale multiple intents datasets and we hope the release of it would push forward the research of multi-intent SLU. In addition, our framework achieves state-of-the-art performance on two public single-intent datasets including ATIS (Tur and De Mori, 2011) and SNIPS (Coucke et al., 2018), which further verifies the generalization of the proposed model.

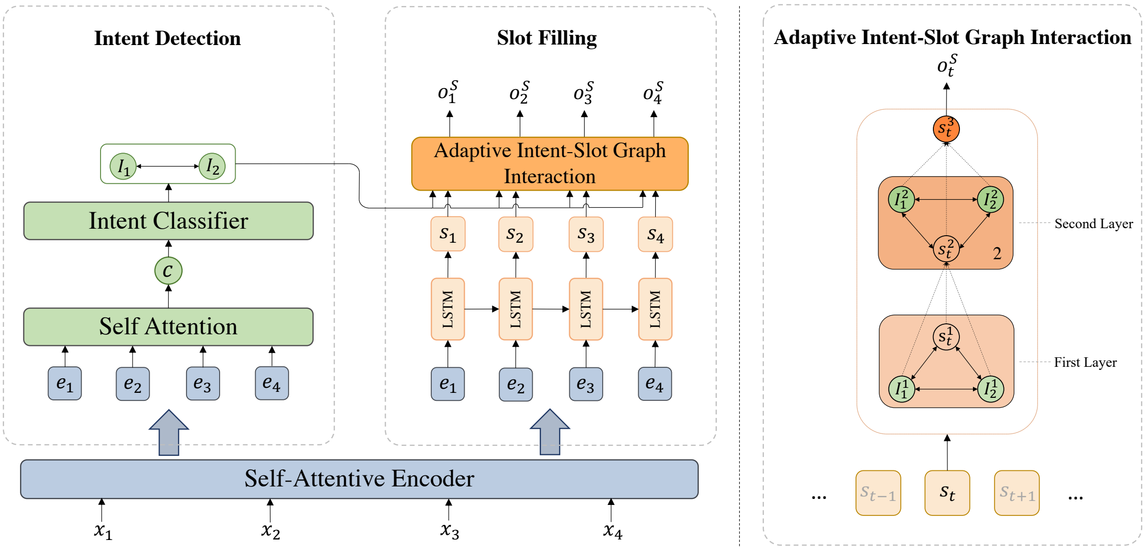
To facilitate future research in this area, all datasets and codes are publicly available at https://github.com/LooperXX/AGIF.

2 Approach

The architecture of our framework is demonstrated in Figure 2, which consists of a shared encoder, an adaptive intent-slot graph interaction layer and two separate decoders. First, the encoder (§2.1) uses a shared self-attentive encoder to represent an utterance, which can grasp the shared information between intent detection and slot filling. Then, the intent detection decoder (§2.2) performs the multi-label classification to detect multiple intents. Finally, we introduce the adaptive intent-slot graph interaction layer (§2.3) to explicitly leverage the multiple intents information for guiding slot prediction. Both intent detection and slot filling are optimized simultaneously via a multi-task learning scheme.

2.1 Self-Attentive Encoder

In the self-attentive encoder, following Qin et al. (2019), we use BiLSTM with the self-attention mechanism to leverage both advantages of temporal features within word orders and contextual information.



(a)

(b)

Figure 2: The overflow of model architecture (a) and adaptive intent-slot graph interaction module (b).

Bidirectional LSTM A bidirectional LSTM (BiLSTM) (Hochreiter and Schmidhuber, 1997) consists of two LSTM layers. For the input sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ (T is the number of tokens in the input utterance), the BiLSTM reads it forwardly from $\boldsymbol{x}_1$ to $\boldsymbol{x}_T$ and backwardly from $\boldsymbol{x}_T$ to $\boldsymbol{x}_1$ to produce a series of context-sensitive hidden states $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T\}$.

Self-Attention We follow Vaswani et al. (2017) to use a self-attention mechanism over word embedding to capture context-aware features. We first map the matrix of input vectors $\boldsymbol{X} \in \mathbb{R}^{T \times d}$ (d represents the mapped dimension) to queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ matrices by using different linear projections parameters $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$. Attention weight is computed by dot product between $\boldsymbol{Q}$, $\boldsymbol{K}$ and the self-attention output $\boldsymbol{A} \in \mathbb{R}^{T \times d}$ is a weighted sum of values:

$$\boldsymbol{A} = \operatorname{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V}\,, \tag{1}$$

where d_k denotes the dimension of keys.

We concatenate these two representations as the final encoding representation:

$$\boldsymbol{E} = [\boldsymbol{H} \,||\, \boldsymbol{A}]\,, \tag{2}$$

where $\boldsymbol{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T\} \in \mathbb{R}^{T \times 2d}$ and $||$ is concatenation operation.

2.2 Intent Detection Decoder

We follow Gangadharaiah and Narayanaswamy (2019) to perform multiple intent detection as the multi-label classification problem. We compute the utterance context vector over $\boldsymbol{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T\} \in \mathbb{R}^{T \times 2d}$. In our case, we use a self-attention module (Zhong et al., 2018; Goo et al., 2018) to capture relevant context:

$$p_t = \operatorname{softmax}(\boldsymbol{w}_e\, \boldsymbol{e}_t + \boldsymbol{b})\,, \tag{3}$$

$$\boldsymbol{c} = \sum_t p_t \boldsymbol{e}_t\,, \tag{4}$$

where $\boldsymbol{w}_e \in \mathbb{R}^{1 \times 2d}$ is the trainable parameters, p_t is corresponding normalized self-attention score.

$\boldsymbol{c}$ is the weighted sum of each element $\boldsymbol{e}_t$ and utilized for intent detection:

$$\boldsymbol{y}^I = \sigma(\boldsymbol{W}_i(\operatorname{LeakyReLU}(\boldsymbol{W}_c\, \boldsymbol{c} + \boldsymbol{b}_c)) + \boldsymbol{b}_i)\,, \tag{5}$$

where $\boldsymbol{W}_i, \boldsymbol{W}_c$ are trainable parameters of the intent decoder, $\boldsymbol{y}^I = \{y_1^I, \ldots, y_{N_I}^I\}$ is the intent output of the utterance and N_I is the number of single intent labels. σ represents the sigmoid activation function.

During inference, we predict intents $\boldsymbol{I} = \{I_1, \ldots, I_n\}$ and I_i represents probability $y_{I_i}^I$ greater than t_u, where $0 < t_u < 1.0$ is a hyper-parameter tuned using the validation set.[1] For example, if the $\boldsymbol{y}^I = \{0.9, 0.3, 0.6, 0.7, 0.2\}$ and the t_u is 0.5, we predict intents $\boldsymbol{I} = \{1, 3, 4\}$.

2.3 Adaptive Intent-Slot Graph Interaction for Slot Filling

In this paper, one of the core contribution is adaptively leveraging multiple intents to guide the slot

[1] In our experiments, we set t_u as 0.5.

prediction, encouraging each token to capture the corresponding relevant intent information. In particular, we adopt the graph attention network (GAT) (Veličković et al., 2017) to model the interaction between intents and slot at the token-level.

In this section, we first describe the vanilla graph attention network. Then, we show how to directly leverage multiple intents information for slot prediction with the adaptive intent-slot graph interaction layer.

Vanilla Graph Attention Network For a given graph with N nodes, one-layer GAT take the initial node features $\tilde{\boldsymbol{H}} = \{\tilde{\boldsymbol{h}}_1, \ldots, \tilde{\boldsymbol{h}}_N\}, \tilde{\boldsymbol{h}}_n \in \mathbb{R}^F$ as input, aiming at producing more abstract representation, $\tilde{\boldsymbol{H}}^{'} = \{\tilde{\boldsymbol{h}}_1^{'}, \ldots, \tilde{\boldsymbol{h}}_N^{'}\}, \tilde{\boldsymbol{h}}_n^{'} \in \mathbb{R}^{F'}$, as its output. The graph attention operated on the node representation can be written as:

$$\begin{aligned} \mathcal{F}(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j) &= \text{LeakyReLU}\left(\mathbf{a}^\top [\boldsymbol{W}_h \tilde{\boldsymbol{h}}_i \| \boldsymbol{W}_h \tilde{\boldsymbol{h}}_j]\right), \\ \alpha_{ij} &= \frac{\exp(\mathcal{F}(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j))}{\sum_{j' \in \mathcal{N}_i} \exp\left(\mathcal{F}(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_{j'})\right)}, \\ \tilde{\boldsymbol{h}}_i^{'} &= \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{W}_h \tilde{\boldsymbol{h}}_j\Big), \end{aligned} \tag{6}$$

where $\mathcal{N}_i$ is the first-order neighbors of node i (including i) in the graph, $\boldsymbol{W}_h \in \mathbb{R}^{F' \times F}$ and $\mathbf{a} \in \mathbb{R}^{2F'}$ is the trainable weight matrix, α_{ij} is the normalized attention weight denoting the importance of each $\tilde{\boldsymbol{h}}_j$ to $\tilde{\boldsymbol{h}}_i$ and σ represents the nonlinearity activation function.

GAT inject the graph structure into the mechanism by performing *masked attention*, i.e, GAT only compute $\mathcal{F}(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j)$ for nodes $j \in \mathcal{N}_i$. To stabilize the learning process of self-attention, GAT extend the above mechanism to employ *multi-head attention* from Vaswani et al. (2017):

$$\tilde{\boldsymbol{h}}_i^{'} = \overset{K}{\underset{k=1}{||}} \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \boldsymbol{W}_h^k \tilde{\boldsymbol{h}}_j\Big), \tag{7}$$

where α_{ij}^k is the normalized attention weight computed by the k-th function $\mathcal{F}_k$, $||$ is concatenation operation and K is the number of heads. Thus, the output $\tilde{\boldsymbol{h}}_n^{'}$ will consists of KF' features in the middle layers and the final prediction layer will employ averaging instead of concatenation to get the final prediction results.

Adaptive Intent-Slot Graph Interaction for Slot Prediction We use a unidirectional LSTM as the slot filling decoder. At each decoding step t, the decoder state $\boldsymbol{s}_t$ is calculated by previous decoder state $\boldsymbol{s}_{t-1}$, the previous emitted slot label distribution $\boldsymbol{y}_{t-1}^S$ and the aligned encoder hidden state $\boldsymbol{e}_t$:

$$\boldsymbol{s}_t = \text{LSTM}\left(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}^S, \boldsymbol{e}_t\right). \tag{8}$$

Instead of directly utilizing the $\boldsymbol{s}_t$ to predict the slot label, we build a graphic structure named adaptive intent-slot graph interaction to explicitly leverage multiple intents information to guide the t-th slot prediction. In this graph, the slot hidden state at t time step is $\boldsymbol{s}_t$ and predicted multiple intents information $\boldsymbol{I} = \{I_1, \ldots, I_n\}$, where n denotes the number of predicted intents, are used as the initialized representations at t time step $\tilde{\boldsymbol{H}}^{[0,t]} = \{\boldsymbol{s}_t, \phi^{emb}(I_1), \ldots, \phi^{emb}(I_n)\} \in \mathbb{R}^{(n+1) \times d}$, where d represents the dimension of vertices representation and $\phi^{emb}(\cdot)$ represents the embedding matrix of intents. In addition, the predicted intents are connected to each other to consider their mutual interaction because all of them express the same utterance's intent.

For convenience, we use $\tilde{\boldsymbol{h}}_i^{[l,t]}$ to represent node i in the l-th layer of the graph consisting of the decoder state node and predicted intent nodes at t time step. $\tilde{\boldsymbol{h}}_0^{[l,t]}$ is the slot hidden state representation in the l-th layer. To explicitly leverage the multiple intents information, the slot hidden state node is directly connected to all predicted intents and the slot node representation in the l-th layer can be calculated as:

$$\tilde{\boldsymbol{h}}_i^{[l,t]} = \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{[l,t]} \boldsymbol{W}_h^{[l]} \tilde{\boldsymbol{h}}_j^{[l-1,t]}\Big), \tag{9}$$

where $\mathcal{N}_i$ represents the first-order neighbors of node i, *i.e.*, the decoder state node and the predicted intent nodes, and the update process of all node representations can be calculated by Equation 6, 7 and 9.

With L-layer adaptive intent-slot graph interaction, we obtain the final slot hidden state representation $\tilde{\boldsymbol{h}}_0^{[L,t]}$ at t time step, which adaptively capture important intents information at token-level. The representation $\tilde{\boldsymbol{h}}_0^{[L,t]}$ is utilized for slot filling:

$$\boldsymbol{y}_t^S = \text{softmax}\left(\boldsymbol{W}_s \tilde{\boldsymbol{h}}_0^{[L,t]}\right), \tag{10}$$

$$o_t^S = \arg\max(\boldsymbol{y}_t^S), \tag{11}$$

where o_t^S is the predicted slot label of the t-th word in the utterance.

2.4 Multi-Task Training

Following Qin et al. (2020), we adopt a joint model to consider the two tasks and update parameters by joint optimizing. The intent detection objective is:

$$\mathcal{L}_1 \triangleq -\sum_{k=1}^{N_I} \left(\hat{y}_k^I \log\left(y_k^I\right) + \left(1-\hat{y}_k^I\right) \log\left(1-y_k^I\right)\right). \tag{12}$$

Similarly, the slot filling task objective is defined as:

$$\mathcal{L}_2 \triangleq -\sum_{i=1}^{M}\sum_{j=1}^{N_S} \hat{y}_i^{(j,S)} \log\left(y_i^{(j,S)}\right), \tag{13}$$

where N_I is the number of single intent labels, N_S is the number of slot labels and M is the number of words in an utterance.

The final joint objective is formulated as:

$$\mathcal{L} = \alpha\mathcal{L}_1 + (1-\alpha)\mathcal{L}_2, \tag{14}$$

where α is hyper-parameter.

3 Experiments

3.1 Datasets

Multiple Intent Datasets We conduct experiments on the benchmark DSTC4 (Kim et al., 2017b), which is human-human multi-turn dialogues. We adopt the same dataset partition in the DSTC4 main task and we regard its speech act attributes as intents.[2] It has 12,759 utterances for training, 4,812 utterances for validation and 7,848 utterances for testing.

To verify the generalization of the proposed model, we construct the multi-intent SLU dataset, MixSNIPS. MixSNIPS dataset is collected from the Snips personal voice assistant (Coucke et al., 2018) by using conjunctions, *e.g.*, “and”, to connect sentences with different intents and ensure that the ratio of sentences has 1-3 intents is $[0.3, 0.5, 0.2]$. Finally, we get the 45,000 utterances for training, 2,500 utterances for validation and 2500 utterances for testing on the MixSNIPS dataset. Similarly, we construct another multi-intent SLU dataset, MixATIS, from the ATIS dataset (Hemphill et al., 1990). There are 18,000 utterances for training, 1,000 utterances for validation and 1,000 utterances for testing. The constructed datasets have been released for future research.

Single Intent Datasets In addition, we also conduct experiments on two public benchmark single-intent datasets to validate the efficiency of our proposed model. One is the ATIS dataset (Hemphill et al., 1990) and the other is SNIPS dataset (Coucke et al., 2018), which are widely used as benchmark in SLU research. Both datasets follow the same format and partition as in Goo et al. (2018) and Qin et al. (2019).

3.2 Experimental Settings

The self-attentive encoder hidden units is 256 in all datasets. ℓ_2 regularization is 1×10^{-6} and dropout rate is 0.4 for reducing overfitting. We use Adam (Kingma and Ba, 2014) to optimize the parameters in our model and adopted the suggested hyper-parameters for optimization. The graph layer number is 3 for DSTC4 dataset and 2 for the other datasets. For all the experiments, we select the model which works the best on the dev set and then evaluate it on the test set. All experiments are conducted at TITAN Xp and GeForce RTX 2080Ti. The epoch number is 50 for MixSNIPS and 100 for MixATIS and DSTC4.

3.3 Baselines

We first compare our model with the existing state-of-the-art multi-intent SLU baseline:

Joint Multiple ID-SF. Gangadharaiah and Narayanaswamy (2019) proposes a multi-task framework with the slot-gated mechanism for multiple intent detection and slot filling.

Then, we compare our framework with the existing state-of-the-art single-intent SLU:

1) **Attention BiRNN.** Liu and Lane (2016) propose an alignment-based RNN with the attention mechanism, which implicitly learns the relationship between slot and intent.

2) **Slot-Gated Atten.** Goo et al. (2018) proposes a slot-gated joint model to explicitly consider the correlation between slot filling and intent detection.

3) **Bi-Model.** Wang et al. (2018) proposes the Bi-model to consider the cross-impact between the intent detection and slot filling.

4) **SF-ID Network.** Haihong et al. (2019) proposes an SF-ID network to establish direct connections for the slot filling and intent detection to help them promote each other mutually.

5) **Stack-Propagation.** Qin et al. (2019) adopts a joint model with Stack-Propagation to capture the intent semantic knowledge and perform the token-level intent detection to further alleviate the error

[2] The official DSTC4 pilot tasks' Handbook http://www.colips.org/workshop/dstc4/DSTC4_pilot_tasks.pdf

Model	MixATIS				MixSNIPS			
	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)
Attention BiRNN	86.6	-	71.6	38.7	89.4	-	94.1	62.2
Slot-Gated	**88.1**	-	65.7	38.9	87.8	-	96.0	56.5
Slot-gated Intent	86.7	-	66.2	39.6	87.9	-	94.2	57.6
Bi-Model	85.5	-	72.3	39.1	86.8	-	95.3	53.9
SF-ID	87.7	-	63.7	36.2	89.6	-	96.3	59.3
Stack-Propagation (*concatenation*)	86.6	-	**76.0**	42.8	93.9	-	96.4	75.5
Stack-Propagation (*sigmoid-decoder*)	87.4	79.0	71.9	41.0	93.2	97.6	94.6	71.9
Joint Multiple ID-SF	87.5	80.6	73.1	38.1	91.0	98.2	95.7	66.6
AGIF	**88.1**	**81.2***	75.8	**44.5***	**94.5***	**98.6***	**96.5***	**76.4***

Table 1: Slot filling and intent detection results on two self-constructed multi-intent datasets. The numbers with * indicate that the improvement of our model over all the compared baselines is statistically significant with $p < 0.05$ under the t-test.

Model	DSTC4			
	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)
Attention BiRNN	44.0	-	42.1	32.6
Slot-Gated	45.0	-	42.5	32.5
Slot-gated Intent	50.2	-	40.6	31.7
Bi-Model	44.6	-	41.3	30.5
SF-ID	51.4	-	41.8	33.0
Stack-Propagation (1)	52.8	-	44.9	34.6
Stack-Propagation (2)	51.9	39.2	39.2	30.5
Joint Multiple ID-SF	48.0	37.5	39.0	29.4
AGIF	**53.9**	**40.0**	**46.1**	**35.2**

Table 2: Slot filling and intent detection results on the DSTC4 dataset. Stack-Propagation (1) denotes the Stack-Propagation (*concatenation*) version and Stack-Propagation (2) denotes the Stack-Propagation (*sigmoid-decoder*) version.

propagation. This model achieves the state-of-the-art performance.

To enable single-intent SLU baselines can handle the multi-intent utterances, we follow Gangadharaiah and Narayanaswamy (2019) to connect them with # to get the single multi-intent label for a fair comparison, we name it as *concatenation* version. To further verify the effectiveness of our framework, we change the state-of-the-art baseline *Stack-Propagation* to directly predict the multi-intent label by changing the inten decoder with replacing softmax as sigmoid and using binary cross-entropy loss. We refer it as the *sigmoid-decoder*.

For the *Attention BiRNN*, *Slot-Gated Atten*, *SF-ID Network* and *Stack-Propagation*, we run their official source code to obtain the results. For the *Bi-Model* and *Joint Multiple ID-SF*, we re-implemented the models and obtained the results on the same datasets because the original paper did not release their codes.

3.4 Main Results

Following Goo et al. (2018) and Qin et al. (2019), we evaluate the performance of slot filling using F1 score, intent prediction using accuracy and macro F1 score, the sentence-level semantic frame parsing using overall accuracy which represents all metrics are right in an utterance. Table 1 shows the experiment results of the proposed models on the MixATIS and MixSNIPS datasets.

From the results, we have three observations:

1) Our framework outperforms *Joint Multiple ID-SF* baseline by a large margin and achieves state-of-the-art performance. On the MixATIS dataset, we achieve 0.6% improvement on Slot (F1) score, 0.6% improvement on Intent (F1), 2.7% improvement on Intent (Acc). On the MixSNIPS dataset, we achieve 3.5% improvement on Slot (F1) score, 0.4% improvement on Intent (F1), 0.8% improvement on Intent (Acc). This indicates that our adaptive intent-slot graph interaction successfully incorporates relevant intent information to improve slot prediction. In addition, we obtain 6.4% improvement and 9.8% improvement on Overall (Acc) on MixATIS and MixSNIPS dataset, respectively. We attribute this to the fact that our adaptive intent-slot graph interaction mechanism can better help grasp the relationship between the intent and slots and improve the whole SLU.

2) The *concatenation* outperforms the *sigmoid-decoder* version, this is because *concatenation* can greatly reduce the multi-intent search space, which makes it easier for single intent systems to predict multiple intents. For example, on the ATIS dataset, there exist 17 single intents and 4 combined multi-intent in the training data. The multi-intent systems make a binary prediction at each intent while the *concatenation* model predicts the limited combined intent search space $(17 + 4)$.

3) Though facing the difficulty of multi-intent prediction, our framework outperforms the state-of-the-art single-intent model (*Stack-Propagation (concatenation)*), which further proves the proposed token-level adaptive graph interaction layer

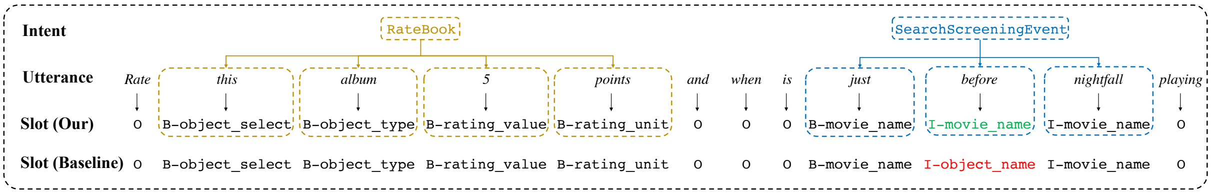


Figure 3: A case study between our model and *Joint Multiple ID-SF*. The green slot is correct while the red one is wrong. Better viewed in color.

.

Model	MixSNIPS			
	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)
Vanilla Attention Interaction	93.8	98.0	95.2	74.0
GCN-based Interaction	93.3	98.3	96.0	72.7
Sentence-Level Augmented	93.8	98.1	95.7	73.9
+ More Parameters	94.1	98.6	96.6	73.6
AGIF	**94.5**	**98.6**	**96.5**	**76.4**

Table 3: Ablation Study on MixSNIPS Datasets.

can improve the SLU performance.

3.5 Analysis

3.5.1 Performance on the DSTC4 dataset

To further analyze the performance of the AGIF model, we conduct experiments on the real-world multi-intent SLU dataset, DSTC4. The results are shown in Table 2. From the results, we achieve 5.9% improvement on Slot (F1) score, 2.5% improvement on Intent (F1), 7.1% improvement on Intent (Acc) and 5.8% improvement on Overall (Acc) compared with *Joint Multiple ID-SF*. This further proves that our adaptive intent-slot graph interaction could aggregate the pertinent intent information to enhance the token-level slot prediction.

3.5.2 Effectiveness of Intent-Slot Graph Interaction Mechanism

- **Graph Attention Mechanism vs. Vanilla Attention Mechanism** Instead of adopting the GAT to model the interaction between the predicted intents and slot, we utilize the attention mechanism to incorporate the intents information for slot filling at the token-level. We name it as *Vanilla Attention Interaction*. We first use the hidden state of slot filling decoder as the query to attend to the intent embedding to obtain the context intent vector, and then we sum the vector and the hidden state of slot filling decoder to get the final slot prediction. The results are shown in *Vanilla Attention Interaction* row in Table 3, we observe the overall performance drops 2.4% on the MixSNIPS dataset. We attribute it to the fact that the multi-layer graph attention network can automatically capture relevant intents information and better aggregate intents information for each token slot prediction.

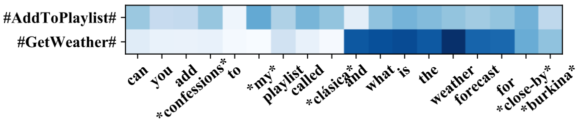


Figure 4: Visualization. Y-axis is the predicted intents and X-axis is the input utterance where slot tokens are surrounded by $*$. For each column, the darker the color, the more relevant they are.

- **Graph Attention Mechanism vs. Graph Convolution Mechanism** We replace the graph attention layer with the graph convolution layer and keep other components unchanged. We refer to it as *GCN-based Interaction*. The results are shown in *GCN-based Interaction* row in Table 3, we observe the performance drops in all metrics in the MixSNIPS dataset. We suggested that *GCN-based Interaction* cannot adaptively attribute different weights to each node in the intent-slot graph while our graph attention mechanism can automatically filter irrelevant intent information for each token.

3.5.3 Effectiveness of Adaptive Intent-Slot Interaction Mechanism

- **Adaptive Interaction Mechanism vs. Sentence-Level Augmented Mechanism** We first conduct experiments by statically providing the same intent information for all tokens slot prediction where we sum the predicted intent embeddings and directly add it to the hidden state of slot filling decoder. We refer to it as *sentence-level augmented*. The result is shown in Table 3. We can observe that if we only provide overall intent information for slot filling, we obtain the worse results, which demonstrates the

Model	ATIS				SNIPS			
	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (F1)	Intent (Acc)	Overall (Acc)
SF-ID	95.6	-	96.6	86	90.5	-	97	78.4
Stack-Propagation	95.9	-	96.9	86.5	94.2	-	98.0	86.9
Joint Multiple ID-SF	94.2	-	95.4	-	88.0	-	97.2	-
AGIF	**96.0**	**80.2**	**97.1**	**87.2**	**94.8**	**98.3**	**98.1**	**87.3**

Table 4: Slot filling and intent detection results on two single-intent datasets.

effectiveness of adaptively incorporating intent information at the token-level. We believe the reason is that providing the same intents for all tokens can cause the ambiguity where each token is hard to extract the relevant intent information while our adaptive intent interaction mechanism can achieve the fine-grained intent interaction and capture the related intent information to guide the slot prediction.

A natural question that raised is whether the more parameters involved by AGIF contribute to the final performance. To verify that the proposed adaptive interaction mechanism rather than the added parameters works, for *sentence-level augmented mechanism* model, we apply multiple LSTM layers (2-layers) to slot filling decoder and we name it as *more parameters*. The results in Table 3 show that our framework outperforms the *more parameters* model in overall accuracy, which verifies that the improvements comes from the proposed adaptive intent-slot interaction mechanism rather than the involved parameters.

- **Qualitative Analysis.** We provide a case study to intuitively understand the token-level adaptive intent-slot interaction mechanism. As shown in Figure 3, AGIF predicts “I-movie_name” correctly for the slot label of “*before*” while *Joint Multiple ID-SF* predicts it as “I-object_name” incorrectly. We observed that “I-object_name” doesn’t belong to the intent “SearchScreeningEvent” but to the intent “RateBook”. We attribute it to the reason that each token is guided with the same complex intents information making it incorrectly and confusedly capture the information of the other intent “RateBook”. In contrast, our adaptive graph interaction mechanism can offer fine-grained intent information integration for token-level slot filling to predict the slot label correctly.

3.5.4 Visualization

With the attempt to better understand what the adaptive intent-slot graph interaction layer has learned, we visualize the intent attention weights of slot filling hidden states node in the output head of the adaptive intent-slot graph interaction layer, which is shown in Figure 4. Based on the utterance “*can you add confessions to my playlist called clásica and what is the weather forecast for close-by burkina*” and the intents “AddToPlaylist” and “GetWeather”, we can clearly see the attention weights successfully focus on the correct intent, which means our graph interaction layer can learn to incorporate the correlated intent information at each slot. More specifically, our model properly aggregates the corresponding “AddToPlaylist” intent information at slots “*confessions, my, clásica*” and “GetWeather” intent information at slots“*close-by burkina*”.

3.5.5 Evaluation on the Single-Intent Datasets

We conduct experiments on two public single-intent benchmarks to evaluate the generalizability of our framework. We compare our model with the single-intent state-of-the-art models including *SF-ID*, *Stack-Propagation* and multi-intent model including *Joint Multiple ID-SF*. Table 4 shows the experiment results of the proposed models on the ATIS and SNIPS datasets. From the table, we can see that our model outperforms all the compared baselines and achieves state-of-the-art performance. This demonstrates the generalizability and effectiveness of our framework whether handling multi-intent or single-intent SLU.

4 Related Work

Intent Detection Intent detection is formulated as an utterance classification problem. Different classification methods, such as support vector machine (SVM) and RNN (Haffner et al., 2003; Sarikaya et al., 2011), have been proposed to solve it. Xia et al. (2018) adopts a capsule-based neural network with self-attention for intent detection.

However, the above models mainly focus on the single intent scenario, which can not handle the complex multiple intent scenario. Xu and Sarikaya (2013b) and Kim et al. (2017a) explore the complex scenario, where multiple intents are assigned to a user's utterance. Xu and Sarikaya (2013b) use log-linear models to achieve this, while we use neural network models. Compared with their work, we jointly perform multi-label intent detection and slot prediction, while they only consider the subtask intent detection.

Slot Filling Slot filling can be treated as a sequence labeling task. The popular approaches are conditional random fields (CRF) (Raymond and Riccardi, 2007) and recurrent neural networks (RNN) (Xu and Sarikaya, 2013a; Yao et al., 2014). Recently, Shen et al. (2018) and Tan et al. (2018) introduce the self-attention mechanism for CRF-free sequential labeling.

Joint Model To consider the high correlation between intent and slots, many joint models (Goo et al., 2018; Li et al., 2018; Xia et al., 2018; E et al., 2019; Liu et al., 2019; Qin et al., 2019) are proposed to solve two tasks. Goo et al. (2018); Li et al. (2018); Zhang et al. (2019) propose to utilize the intent information to guide the slot filling. Qin et al. (2019) further utilize a stack-propagation framework for better leveraging intent semantic information to guide the slot filling, which achieves the state-of-the-art performance. Wang et al. (2018) and E et al. (2019) consider the cross-impact between the slot and intents. Our framework follows those state-of-the-art joint model paradigm, and further focus on the multiple intents scenario while the above joint models do not consider. Recently, Gangadharaiah and Narayanaswamy (2019) propose a joint model to consider the multiple intent detection and slot filling simultaneously where they explicitly leverage overall intent information with the gate mechanism to guide all tokens slot prediction. Compared with this work, the main differences are as following: 1) Our framework exploits a fine-grained intent information transfer with a unified graph interaction architecture while their work simply incorporates the same intents information for all tokens slot prediction. 2) As far as we know, their corpus and code are not distributed, which makes it hard to follow. In contrast, we empirically construct two large-scale multi-intent SLU datasets where all datasets and code have been released. We hope it would push forward the research of multi-intent SLU.

5 Conclusion

In our paper, we propose a token-level adaptive graph-interactive framework to model the interaction between multiple intents and slot at each token, which can make a fine-grained intent information transfer for slot prediction. To our best of knowledge, this is the first work to explore fine-grained intents information transfer in multi-intent SLU. In addition, we release two multi-intent datasets and hope it can push forward the research this area. Experiments on four datasets show the effectiveness of the proposed models and achieve state-of-the-art performance.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011 and 61772153.

References

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proc. of ACL*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proc. of NAACL*, pages 564–569.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proc. of NAACL*.

Patrick Haffner, Gokhan Tur, and Jerry H Wright. 2003. Optimizing svms for complex call classification. In *In Proc. of ICASSP*.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proc. of ACL*, pages 5467–5471.

Charles T Hemphill, John J Godfrey, and George R Doddington. 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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8).

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017a. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76(9):11377–11390.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, and Matthew Henderson. 2017b. The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proc. of EMNLP*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. Cm-net: A novel collaborative memory network for spoken language understanding. *arXiv preprint arXiv:1909.06937*.

Libo Qin, Wanxiang Che, Yangming Li, Minheng Ni, and Ting Liu. 2020. Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. In *AAAI*, pages 8665–8672.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proc. of EMNLP*.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*.

Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *ICASSP*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proc. of NAACL*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proc. of AAAI*.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proc. of AAAI*.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł *ukasz* Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bimodel based rnn semantic frame parsing model for intent detection and slot filling. In *Proc. of ACL*.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proc. of EMNLP*, pages 3090–3099, Brussels, Belgium. Association for Computational Linguistics.

Puyang Xu and Ruhi Sarikaya. 2013a. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*.

Puyang Xu and Ruhi Sarikaya. 2013b. Exploiting shared information for multi-intent natural language sentence classification. In *INTERSPEECH*.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *SLT*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proc. of ACL*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proc. of ACL*.