Enhancing Joint Multiple Intent Detection and Slot Filling with Global Intent-Slot Co-occurrence

Mengxiao Song, Bowen Yu, Quangang Li*,
Yubin Wang, Tingwen Liu, Hongbo Xu
Institute of Information Engineering, Chinese Academy of Sciences. Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences. Beijing, China
{songmengxiao,yubowen,liquangang}@iie.ac.cn
{wangyubin,liutingwen,hbxu}@iie.ac.cn

Abstract

Multi-intent detection and slot filling joint model attracts more and more attention since it can handle multi-intent utterances, which is closer to complex real-world scenarios. Most existing joint models rely entirely on the training procedure to obtain the implicit correlation between intents and slots. However, they ignore the fact that leveraging the rich global knowledge in the corpus can determine the intuitive and explicit correlation between intents and slots. In this paper, we aim to make full use of the statistical co-occurrence frequency between intents and slots as prior knowledge to enhance joint multiple intent detection and slot filling. To be specific, an intent-slot co-occurrence graph is constructed based on the entire training corpus to globally discover correlation between intents and slots. Based on the global intent-slot co-occurrence, we propose a novel graph neural network to model the interaction between the two subtasks. Experimental results on two public multi-intent datasets demonstrate that our approach outperforms the state-of-the-art models.

1 Introduction

Spoken language understanding (SLU) aims to enable the machine to understand user's utterance, which contains two typical subtasks: intent detection and slot filling (Tur and De Mori, 2011). Intent detection obtains the user's intent from the input utterance and slot filling recognizes entities carrying detailed information of the intent. Gangadharaiah and Narayanaswamy (2019) found that complex scenarios of real-world may often include multiple intents in an utterance. Take an example in Figure 1, the utterance indicates two intents: "GetWeather" and "BookRestaurant". Jeong and Lee (2008) suggested that solving the two subtasks jointly makes better performance because of the correlation between intents and slots (e.g. "BookRestaurant" and B-restaurant_type). Therefore, researchers in recent years (Gangadharaiah and Narayanaswamy, 2019; Qin et al., 2020, 2021b) pay more attention to joint multiple intent detection and slot filling.

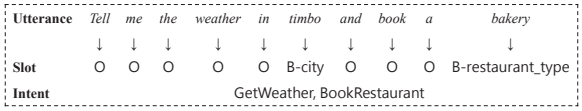

Figure 1: An example of joint multiple intent detection and slot filling for an utterance.

Gangadharaiah and Narayanaswamy (2019) first tried to explore the joint model for multiple intent detection and slot filling based on attention with a slot-gated mechanism to model dependencies between intents and slots. Qin et al. (2020) proposed an adaptive graph-interactive framework (AGIF) where the core is an adaptive intent-slot interaction multilayer graph attention network. It makes it possible for each token to capture different relevant intent information, thus enabling fine-grained integration of multiple intents. On the basis of AGIF, Qin et al. (2021b) proposed a global-locally graph-interaction network (GL-GIN), which builds a local slot-aware graph and a global intent-slot graph for each utterance to model slot dependency and intent-slot interaction. Nevertheless, these works rely entirely on model training to obtain implicit intent-slot correlation for interaction between the two subtasks. And existing works often construct graphs for each individual utterance, which neglects global statistical knowledge from the entire corpus. The simplified example is illustrated in Figure 2(a).

To overcome the above limitations, inspired by the success of leveraging global knowledge from training sets to help model optimization in multi-label classification (Zhang et al., 2018; Chen et al., 2019; Ma et al., 2021), we decide to explicitly leverage the intent-slot correlation summarized from the

*Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7967–7977
December 7-11, 2022

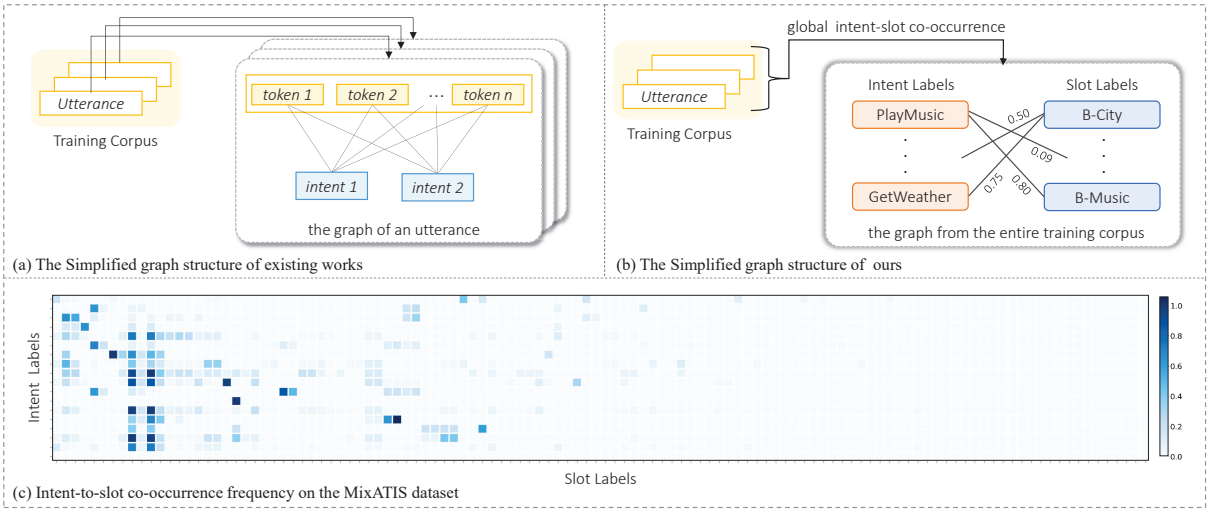

Figure 2: Graph structure comparison. (a): The simplified graph structure of existing works. (b): The simplified graph structure our global intent-slot co-occurrence graph. (c): The visualization of the intent-to-slot co-occurrence frequency counted on the MixATIS training set. Darker color represents higher co-occurrence frequency.

entire training corpus. By counting co-occurrence numbers between intents and slots in utterances and calculating the frequency, we visualize the statistical results on the MixATIS (which is a public multi-intent utterance dataset) in Figure 2(c). We can clearly see that there are significant differences in co-occurrence frequency between different intents and slots. Based on the observation, we propose to regard the co-occurrence frequency as prior knowledge for guiding the model to mine the correlation between intents and slots.

Methodologically, we devise an intent-slot graph with intent and slot labels as vertices, and the statistical co-occurrence frequency between intents and slots across the entire training corpus is formulated as weighted edges. Figure 2(b) shows the simplified example of our graph. The weight between each intent and each slot can be regarded as the correlation degree of them. Then, we apply the graph convolutional network (GCN) to the intent-slot graph to propagate information and update vertices embeddings. In this way, integrated global knowledge from the corpus is capable of helping the model capture explicit correlation between intents and slots, thereby facilitating interactions between the two subtasks to boost performance. Moreover, we explore several attention and fusion mechanisms to extract label-related information from each utterance and the global training corpus, for the purpose of incorporating semantic information of tokens that play an important role in judging slots and intents, while reducing noise from irrelevant tokens.

We conduct experiments on two public multi-intent utterance datasets. Experimental results show that our approach significantly outperforms the previous state-of-the-art model in the overall accuracy of utterances. And extensive experimental analyses justify that our approach can make better use of the global intent-slot co-occurrence to help the model capture more accurate correlations between intents and slots, so that enhance joint multiple intent detection and slot filling. The source code for this paper can be obtained from https://github.com/smxiao/GISCo.

2 Methodology

Problem Definition Given input utterance $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n]$, where n is the length of the utterance. Joint multiple intent detection and slot filling contains two subtasks: (1) *multiple intent detection* can be seen as a multi-label classification task that predicts the multiple intents $\boldsymbol{O}^I = [\boldsymbol{o}_1^I, ..., \boldsymbol{o}_m^I]$ of the input utterance, where m indicates the number of output intent labels; (2) *slot filling* can be defined as a sequence labeling task that predicts the slot label for each token of the input utterance and output a slot label sequence $\boldsymbol{O}^S = [\boldsymbol{o}_1^S, ..., \boldsymbol{o}_n^S]$.

As shown in Figure 3, our approach is composed of four major components: (1) an utterance encoder; (2) intent and slot label embedder; (3) intent-slot co-occurrence graph network; (4) multi-intent detection and slot filling decoders. And we use a joint training scheme to optimize multiple intent

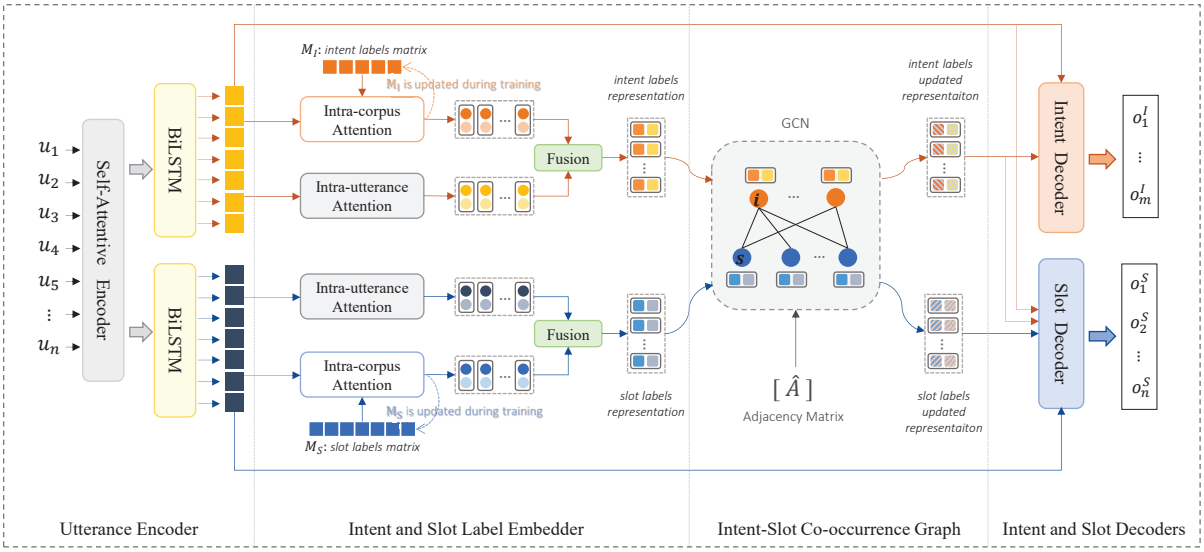

Figure 3: The architecture of our proposed approach. Better viewed in color.

detection and slot filling simultaneously. Next, we will describe each component in detail.

2.1 Utterance Encoder

Following Qin et al. (2020) and Qin et al. (2021b), we leverage task-shared and task-specific encoders to obtain the representation of utterance.

Task-shared Utterance Encoder For the input utterance $\boldsymbol{U}$, we first utilize a self-attentive encoder to obtain the shared utterance representation of two subtasks. Concretely, we first use a Bi-LSTM (Hochreiter and Schmidhuber, 1997) to obtain content representation $\mathbf{C}_1$ with timing information of the input utterance. Then, we employ a self-attention mechanism (Vaswani et al., 2017) over the embedding of the input utterance to capture the context-aware features $\mathbf{C}_2$. After that, we concatenate the output of Bi-LSTM and self-attention as the task-shared utterance representation $\mathbf{E} = \mathbf{C}_1 \oplus \mathbf{C}_2$.

Task-specific Utterance Encoder We utilize two different Bi-LSTMs to capture intent-aware and slot-aware contexts as task-specific features for intent detection and slot filling respectively. More specifically, we feed the task-shared utterance representation $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n]$ into an intent-aware Bi-LSTM, and apply $\mathbf{e}_t^I = \text{BiLSTM}(\mathbf{e}_t, \mathbf{e}_{t-1}^I, \mathbf{e}_{t+1}^I)$ repeatedly to obtain the task-specific representation $\mathbf{E}_I = [\mathbf{e}_1^I, \mathbf{e}_2^I, ..., \mathbf{e}_n^I]$. The slot-aware Bi-LSTM is modeled similarly and outputs $\mathbf{E}_S$.

2.2 Intent and Slot Label Embedder

In this subsection, we describe how to represent intent and slot labels. We devise an intra-utterance attention mechanism and an intra-corpus attention mechanism to collect label features from the input utterance and the training corpus, which are merged together with an adaptive fusion mechanism.

Intra-utterance Attention Mechanism Our motivation comes from the observation that each word in one utterance has a different effect on different labels (Xiao et al., 2019), so we consider extracting label-related semantic components from utterance to represent intent and slot labels. Concretely, we first compute the label-word attention score (Lin et al., 2017). Then, we can capture label-related contextual information from utterance as the representation of the label with the attention score. The process is calculated as follows:

$$\mathbf{B}_\delta = \text{Softmax}(\mathbf{W}_2^\delta \text{tanh}(\mathbf{E}_\delta \mathbf{W}_1^\delta)^{\mathrm{T}})\mathbf{E}_\delta, \quad (1)$$

where $\mathbf{W}_1^\delta \in \mathbb{R}^{d \times d_1}$ and $\mathbf{W}_2^\delta \in \mathbb{R}^{|\delta| \times d_1}$ are the parameters to be trained, $\delta \in \{I, S\}$ (I denotes intent label and S denotes slot label), $|\delta|$ is the number of pre-dedfined labels. $\mathbf{B}_I$ and $\mathbf{B}_S$ are the obtained intent label representation and slot label representation from utterance, respectively.

Intra-corpus Attention Mechanism While the intra-utterance attention mechanism dynamically collects utterance-specific label features from each input utterance, we argue that it ignores the shared features among all the utterances in the training corpus. Thus, we also devise an intra-corpus at-

tention mechanism as a supplement. Specifically, we first randomly initialize the intent and slot label matrices $\mathbf{M}_\delta \in \mathbb{R}^{|\delta| \times d}$ ($\delta \in \{I, S\}$), respectively. Then we update $\mathbf{M}_\delta$ according to its relevance with each utterance in the training corpus as follows:

$$\mathbf{M}_\delta = (\mathbf{M}_\delta \mathbf{E}_\delta^{\mathrm{T}})\mathbf{E}_\delta, \tag{2}$$

where the relevance is calculated by the simple dot product operation. So that, after seeing all the instances in the training corpus, $\mathbf{M}_\delta$ is desired to capture the shared label-specific utterance features. After training, $\mathbf{M}_\delta$ is fixed during inference, while $\mathbf{B}_\delta$ is dynamically generated for each test utterance. This is the most significant difference between them.

Adaptive Fusion Mechanism We utilize an adaptive attention fusion strategy to fully and appropriately fuse $\mathbf{B}_\delta$ and $\mathbf{M}_\delta$. The fusion strategy adaptively extracts a reasonable proportion of information from the two parts to generate the fused label representation. To be specific, we apply fully connected layer to get the proportion of each type of representation in the fusion operation:

$$\boldsymbol{\alpha}_\delta = \frac{\sigma(\mathbf{B}_\delta \mathbf{W}_3^\delta)}{\sigma(\mathbf{B}_\delta \mathbf{W}_3^\delta) + \sigma(\mathbf{M}_\delta \mathbf{W}_4^\delta)}, \tag{3}$$

where $\mathbf{W}_3^\delta$, $\mathbf{W}_4^\delta \in \mathbb{R}^d$ are the parameters to be trained, and σ denotes the Sigmoid function. Once having the fusion weights, we can get the fused label representation as follows:

$$\mathbf{H}_\delta = \boldsymbol{\alpha}_\delta \mathbf{B}_\delta + (\mathbf{1} - \boldsymbol{\alpha}_\delta)\mathbf{M}_\delta. \tag{4}$$

Therefore, the intent labels and slot labels can be embedded into representations $\mathbf{H}_I$ and $\mathbf{H}_S$.

2.3 Intent-Slot Co-occurrence Graph

Graph Construction We construct the intent-slot co-occurrence graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices refer to intent and slot labels, and edges refer to the statistical co-occurrence frequency between intents and slots counted from the training corpus.

Vertex There are two kinds of vertices in the graph: one is the intent label vertex; the other is slot label vertex. The number of vertices is $|I|+|S|$. The initial embeddings of vertices in the graph are produced by $\mathbf{H}_g = [\mathbf{H}_I, \mathbf{H}_S]$.

Edge We define the adjacent matrix of the graph based on the co-occurrence frequency between intents and slots on the training set:

$$\mathbf{A}_{ij} = \begin{cases} \frac{Count((i,j))}{Count(i)}, & \textit{if i and j co-exist,} \\ 0 & \textit{, otherwise,} \end{cases} \tag{5}$$

where i and j are different types of vertices, i.e. if i is an intent vertex then j is a slot vertex, $Count(\cdot)$ denotes the cumulative number of occurrence, and particularly $\mathbf{A}_{ii} = 1$ for self-loop. The shape of $\mathbf{A}$ is $(|I| + |S|) \times (|I| + |S|)$. Then the adjacent matrix $\mathbf{A}$ is normalized by the method in Kipf and Welling (2016) as follows:

$$\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, \tag{6}$$

in which $\mathbf{D}$ is a diagonal degree matrix with entries $\mathbf{D}_{ij} = \sum_j \mathbf{A}_{ij}$.

Graph Network The graph convolutional network (GCN) (Kipf and Welling, 2016) is a convolutional neural network working on a graph, which updates vertex embedding by propagating information between neighboring vertices via adjacent edges. K-layers GCN can aggregate information from K-hops neighbors. Noted that, one layer of GCN is enough for our graph because we only consider the information interaction between intent label and slot label, and they are one-hop neighbors to each other, otherwise irrelevant information may be aggregated. Hence, the interaction between intent and slot labels can be formulated as:

$$\mathbf{H}_g^{'} = \hat{\mathbf{A}} \mathbf{H}_g \mathbf{W}_g, \tag{7}$$

where $\mathbf{W}_g$ is a transformation matrix to be learned. Then the $\mathbf{H}_g^{'}$ is split into $\mathbf{H}_I^g$ and $\mathbf{H}_S^g$ which are the updated intent and slot label representations, respectively.

2.4 Intent and Slot Decoders

Multiple Intent Detection Decoder First, we simply calculate the dot product of intent label representation and utterance representation to obtain their similarity for intent detection. Then, a multi-layer perception is applied to get the predicted probability of each intent label. The process is defined as follows:

$$\mathbf{I} = \sigma(f((\mathbf{H}_I^g \mathbf{E}_I)\mathbf{W}_6^I)\mathbf{W}_5^I), \tag{8}$$

where σ and f denotes the Sigmoid and LeakyReLU activation function respectively.

The above formula is designed for token-level intent detection. To get final sentence-level intent labels, we apply the same voting mechanism as Qin et al. (2021b) to output intents O^I contained in the input utterance:

$$\boldsymbol{O}^I = [\boldsymbol{o}_k^I | (\sum_{t=1}^{n} \mathbb{1}[\mathbf{I}_{t,k} > 0.5]) > \frac{n}{2}], \tag{9}$$

where $\mathbf{I}_{t,k}$ denotes the prediction probability of token t for the intent o_k.

Slot Filling Decoder Inspired by Qin et al. (2021a), in order to make better use of intent information to guide slot filling, we further fuse intent and slot information implicitly via a feed-forward network for slot label prediction. We firstly leverage the attention mechanism to obtain the corresponding intent and slot information from utterance:

$$\mathbf{E}_{\delta}^{'} = (\mathbf{E}_{\delta}\mathbf{H}_{\delta}^{g\mathrm{T}})\mathbf{H}_{\delta}^{g} + \mathbf{E}_{\delta}. \quad (10)$$

Next, the intent and slot information are combined by concatenation $\mathbf{E}_{IS} = \mathbf{E}_{I}^{'} \oplus \mathbf{E}_{S}^{'}$. And we leverage word neighbour features for each token as Zhang and Wang (2016):

$$\hat{\mathbf{e}}_{IS}^{t} = \mathbf{e}_{IS}^{t-1} \oplus \mathbf{e}_{IS}^{t} \oplus \mathbf{e}_{IS}^{t+1}. \quad (11)$$

Then, we integrate the intent information and context features $\hat{\mathbf{E}}_{IS} = [\hat{\mathbf{e}}_{IS}^{1}, ..., \hat{\mathbf{e}}_{IS}^{n}]$ into the slot information by a feed-forward layer, and we add it to $\mathbf{E}_{S}^{'}$ to obtain the enhanced slot information:

$$\mathbf{S}_{u} = \mathrm{ReLU}(\hat{\mathbf{E}}_{IS}\mathbf{W}_{5}^{S})\mathbf{W}_{6}^{S} + \mathbf{E}_{S}^{'}, \quad (12)$$

After that, we apply a multi-layer perception to predict the corresponding slot label for each token:

$$\mathbf{S} = \mathrm{Softmax}(f(\mathbf{S}_{u}\mathbf{W}_{8}^{S})\mathbf{W}_{7}^{S}), \quad (13)$$

where f is the LeakyReLU activation function. Finally, the output $\boldsymbol{O}^{S} = argmax(\mathbf{S})$ is the result of slot filling.

2.5 Joint Training

We apply the joint training to learn parameters. The loss function of multi-intent detection is defined as:

$$CE(\hat{y}, y) = \hat{y}\log(y) + (1-\hat{y})\log(1-y), \quad (14)$$

$$\mathcal{L}_{I} \triangleq -\sum_{k=1}^{n}\sum_{j=1}^{N_I} CE(\hat{o_k}^{(j,I)}, o_k^{(j,I)}), \quad (15)$$

where $\hat{o_k}^{(j,I)}$ is the gold intent label. Similarly, the loss function of slot filling is formulated as:

$$\mathcal{L}_{S} \triangleq -\sum_{k=1}^{n}\sum_{j=1}^{N_S} \hat{o_k}^{(j,S)}, o_k^{(j,S)}, \quad (16)$$

where $\hat{o_k}^{(j,S)}$ is gold slot label. So that, the final joint training objective is as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{I} + (1-\lambda)\mathcal{L}_{S}, \quad (17)$$

in which λ is hyper-parameter.

3 Experiments

3.1 Datasets

We conduct experiments on two public multi-intent utterance datasets: **MixATIS** (Hemphill et al., 1990; Qin et al., 2021b) and **MixSNIPS** (Coucke et al., 2018; Qin et al., 2021b). There are 13,162, 756, 828 utterances for training, validation and testing on the MixATIS dataset, respectively. MixSNIPS includes 39,776, 2,198, 2,199 utterances for training, validation and testing, respectively.

3.2 Experimental Settings

In our experiments, the word embedding is randomly initialized with 128-dimensional word vectors. The dimension of the BiLSTM hidden units is 256. The dimension of the label representation is 384. For the hyper-parameter λ in the loss function of joint training, it is set 0.8 and 0.7 for the MixATIS and MixSNIPS respectively. We set the train batch size to 16. The whole model is trained via AdamW (Loshchilov and Hutter, 2017) with the learning rate being 1e-3, and the weight decay is set 1e-6 and 5e-4 for the MixATIS and MixSNIPS respectively. All experiments are conducted in Tesla T4.

3.3 Baselines

We compare our model with the following baselines: (1) **Attention BiRNN** (Liu and Lane, 2016): It introduces attention mechanism into the bidirectional RNN network for joint intent detection and slot filling. (2) **Slot-Gated Attention** (Goo et al., 2018): It proposes a slot-gated mechanism to learn dependencies between intents and slots. (3) **Bi-Model** (Wang et al., 2018): It considers intent detection and slot filling interacting bidirectionally. (4) **SF-ID Network** (Niu et al., 2019): It utilizes an iterative mechanism to enhance the correlation between intents and slots. (5) **Stack-Propagation** (Qin et al., 2019): It proposes a joint model with stack-propagation which performs the token-level intent detection to guide slot filling. (6) **Joint Multiple ID-SF** (Gangadharaiah and Narayanaswamy, 2019): It applies the slot-gated mechanism to joint multiple intent detection and slot filling. (7) **AGIF** (Qin et al., 2020): It utilizes an adaptive graph interactive framework which can capture the multi-intent information for slot filling. (8) **GL-GIN** (Qin et al., 2021b): It proposes a global-local graph interaction network which is the recent state-of-the-art for the joint task.

Model	Overall(acc)	
	MixATIS	**MixSNIPS**
Attention BiRNN	39.1	59.5
Slot-Gated	35.5	55.4
Bi-Model	34.4	63.4
SF-ID Network	34.9	59.9
Stack-Propagation	40.1	72.9
Joint Multiple ID-SF	36.1	62.9
AGIF	40.8	74.2
GL-GIN	43.5	75.4
Ours	**48.2**	**75.9**

Table 1: Main results on the MixATIS and the MixSNIPS datasets, bold marks highest number among all models. The experimental results of all baseline models are directly cited from Qin et al. (2021b).

3.4 Main Results

Following Goo et al. (2018) and Qin et al. (2021b), we evaluate the performance using the sentence-level overall accuracy (Overall Acc): the prediction of an utterance is correct only when its intents and slots are all correctly predicted.

Table 1 shows experiment results of different models on the two multi-intent datasets. We can see that our approach achieves 4.7% and 0.5% improvements in overall accuracy compared with the best baseline GL-GIN on the MixATIS and MixSNIPS datasets, respectively. One interesting finding is that, our approach achieves comparable results compared with the best baseline on the slot F1 score and intent accuracy which are metrics used to evaluate the two subtasks. Slot F1 score and intent accuracy results are in Appendix A.

We attribute the above gains to our model use of the global statistic result over the entire corpus to provide more information than independent utterance. In this way, taking the intent-slot co-occurrence frequency as prior knowledge can help the neural network capture the more explicit and high-confidence correspondence between intents and slots. In the case that intents (slots) prediction is correct, our approach can better guide the prediction of slots (intents), so that our model can accurately predict all intents and slots of an utterance, thus improving the overall accuracy.

In particular, it is obvious that the improvement on the MixATIS is significant. We intuitively suspect that this is due to the characteristic of the MixATIS dataset: it is collected from an airline travel information system, which has very similar intent labels and slot labels (e.g. `"Flight"` vs. `"Aircraft"`; `"B-airline_code"` vs. `"B-aircraft_code"`). Therefore, the global intent-slot co-occurrence knowledge is more beneficial to distinguishing similar intents and slots. In contrast, the performance improves slightly on the MixSNIPS. We guess that, for this dataset with few label types and obvious label semantic differences (e.g. `"GetWeather"` vs. `"PlayMusic"`), the previous works are also capable of making the model capture partial correlations between intents and slots. Whereas, through leveraging well-informed global statistical knowledge, our method can further help the model to distinguish the correlations between those few labels whose semantic are similar.

3.5 Analysis

3.5.1 Ablation Test

To study the effectiveness of each component in our model, we perform ablation experiments on MixATIS and MixSNIPS datasets.

Effectiveness of the Intent-Slot Co-occurrence Graph Module We remove the global intent-slot co-occurrence in the graph module. That is, all intent vertices and slot vertices are connected to each other with unweighted edges, which is named as *w/o Intent-Slot Co-occurrence* in Table 2. We can observe that overall accuracy drops significantly by 9.2% and 3.4% on MixATIS and MixSNIPS datasets, respectively. This demonstrates that the intent-slot co-occurrence is a very necessary component which captures the correspondence between intent and slot and enhance the joint of two subtasks effectively.

Effectiveness of Attention and Fusion Mechanisms for Label Representation In order to justify the effectiveness of using intra-utterance attention and intra-corpus attention mechanisms together, we use only one of them for intent and slot label representation, which are referred to *w/o Intra-corpus Attention* and *w/o Intra-utterance Self-Attention* in Table 2. We can observe that only using intra-utterance attention for the representation of labels, results in overall accuracy decreases of 7.0% and 8.6% on the two datasets respectively. And only using intra-corpus attention leads to 5.1% and 4.4% overall accuracy drops. Because of intra-utterance leveraging independent utterance context to represent the label, but it overlooks common features of all utterances or tokens with the same intent

Model	Overall(Acc)	
	MixATIS	MixSNIPS
Ours	48.2	75.9
w/o Intent-slot Co-occurrence	39.0	72.5
w/o Adaptive Fusion	47.3	75.2
w/o Intra-Corpus Attention	41.2	67.3
w/o Intra-Utterance Attention	43.1	71.5

Table 2: Ablation test results on the MixATIS and the MixSNIPS datasets.

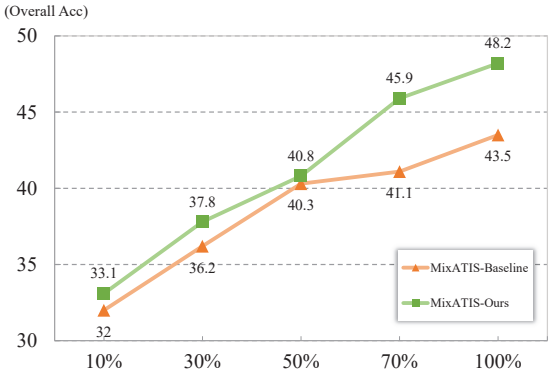

Figure 4: Low-Resource setting experiments results of our approach and GL-GIN on the MixATIS dataset. Better viewed in color.

or slot label in the training corpus. While intra-corpus attention makes up for the defect but ignores the specific context of each utterance. Hence, coupling with them is rational and necessary.

Moreover, we verify the effectiveness of the adaptive fusion mechanism through replacing it with simply adding the two attention mechanisms outputs together, which is named as *w/o Adaptive Fusion Mechanism* in Table 2. We can see that removing the component hurts the results by 0.9% and 0.7% overall accuracy on the two datasets. It justifies adaptive fusion mechanism can indeed help fusing intent (slot) label represent by extracting proper amount of information from independent utterance context and shared label-related features among all utterances in the entire training corpus.

3.5.2 Speedup

We compare the inference speed of our approach with GL-GIN which is the recent fastest baseline. Specifically, we run the model on the MixATIS test set in an epoch, and the batch size is set to 16. The inference latency of our model and the GL-GIN is 1.7 seconds and 4.6 seconds respectively. So our approach implements the $\times$2.7 speedup compared with GL-GIN, which demonstrates that our approach is efficient. The possible reason for this fact that GL-GIN requires the predict results of intent detection to construct the graph on each utterance for slot filling, whereas our approach decodes the two subtasks in parallel.

3.5.3 Low-Resource Setting

We test our approach and the best baseline GL-GIN by varying the ratio of training set from {10, 30, 50, 70, 100} on the MixATIS dataset to compare the overall accuracy of them in low-resource scenarios. The comparison results are shown in Figure 4. We can clearly observe that our approach outperforms the baseline on all five proportions of the MixATIS training set. This demonstrates that the global intent-slot co-occurrence from the entire training corpus is beneficial to boosting the model performance. In addition, it is worth noting that, when the ratio of training set exceeds 70%, the strength of our approach in the overall accuracy is significant. We attribute this to the fact that the richer corpus is more helpful to statistic to determine more accurate correlation between intents and slots, so that the more effective global intent-slot co-occurrence graph structure can be constructed.

3.5.4 Case Study

To further understand how the intent-slot co-occurrence graph works, we provide two cases from the MixATIS and MixSNIPS datasets and show the output results of our approach and of the best baseline GL-GIN. From Figure 5(a), we can see that our model predicts the slot label `"B-aircraft_code"` of token *"j31"* correctly, while GL-GIN predicts it as `"O"` incorrectly. Based on the statistical result of intent-slot co-occurrence frequency on the entire training corpus as prior knowledge, it can be learned that there is strong correlation between the intent `"Quantity"` and the slot `"B-aircraft_code"`. It helps our model predict the token *"j31"* as a slot `"B-aircraft_code"` that carries detailed information of the utterance rather than the label `"O"`. A similar phenomenon can be found in the case from the MixSNIPS dataset as shown in Figure 5(b). The slot label `"B-artist"` and `"B-entity_name"` are sometimes indistinguishable, which can confuses the model and leads to false predictions. However, the statistical co-occurrence fre-

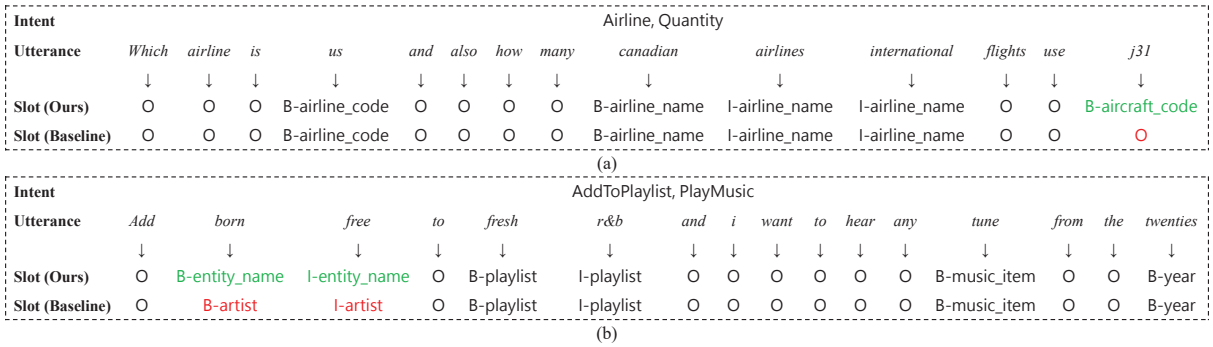

Figure 5: Case study between our model and GL-GIN on the MixATIS dataset (a) and the MixSNIPS dataset (b). The green slot is correct while the red one is wrong.

quency of the intent "AddToPlaylist" and the slot "B-entity_name" is greater than that of "AddToPlaylist" and "B-artist", which can guide our model correctly predict the slot label of token *"born"* and *"free"* as "B-entity_name" and "I-entity_name" respectively.

4 Related Work

The study of joint model for intent detection and slot filling has been central to spoken language understanding in recent years. Because of strong dependency between intent categories and slot labels, jointly solving the two tasks is not only more effective through guiding each other, but also usually requires only one model to be trained and fine-tuned, which reduces the impact of cascading error compared with pipeline models to some extent. Jeong and Lee (2008) proposed the first joint model named triangular chain conditional random fields for slot filling and intent detection, and the model outperforms the most advanced pipeline methods at that time. Guo et al. (2014) applied neural network on utterance parsing tree to solve the joint task. Liu and Lane (2016) explored the strategy of integrating explicit alignment information for slot filling, and further applies attention mechanism to a recurrent neural network (RNN). Goo et al. (2018) introduced the slot-gated mechanism which was the first attempt to explicitly guide slot filling with intent information. Wang et al. (2018) adopt a bidirectional interaction attention module, which means that the interaction is not only from intent to slot, but also from slot to intent oppositely. Zhang et al. (2020) first attempted to use graph neural network to solve the problem of non-parallelization of RNN and inability to capture the remote dependency. Qin et al. (2021a) considered the cross-impact between intent detection and slot filling and proposes a co-interactive transformer for the joint task.

However, the above joint models can mainly deal with single intent utterance, and cannot handle complex cases with multiple intents. Based on the observation that 52% of utterances in an Amazon internal dataset contain more than one intent, Gangadharaiah and Narayanaswamy (2019) first focused on multiple intent scenario. It utilizes attention with slot-gated mechanism to construct a joint framework for multiple intent detection and slot filling. But their model predicts the slot for each token guided with same intent information. For more fine-grained integration of intent information, Qin et al. (2020) applied multi-layer graph attention network for adaptive interaction from intent to slot. Further, aiming at the issue of slow inference speed of existing autoregressive models, Qin et al. (2021b) proposed the first non-autoregressive model for multiple intent detection and slot filling, which is a global-local graph framework to model slot dependency locally and propagate information from multiple intents to slots globally.

In our work, as far as we know, we are the first to consider leveraging the global intent-slot co-occurrence across the entire training corpus to enhance the joint of the two subtasks. Because taking full advantage of the well-informed co-occurrence information between intents and slots from the entire corpus can help model determine more intuitive and accurate correlation between them than constructing a graph for each independent utterance in previous studies.

5 Conclusions

In this paper, we present a new perspective of integrating global intent-slot co-occurrence across

the entire corpus to enhance joint multiple intent detection and slot filling. Specifically, we construct a global intent-slot co-occurrence graph to discover correlation between intents and slots. Then, we combine it with a GCN-based model to guide the information propagation and achieve interaction for the joint task. Besides, we explore several attention mechanisms for dynamically extracting related information from independent utterance context and capturing shared label-specific features among all utterances in the training corpus, then fuse them adaptively to represent intent and slot label. Experimental analyses on two public multi-intent datasets justify that our approach is effective and efficient.

Limitations

The performance of our approach will be limited in the case of insufficient corpus resources. Because our approach requires rich corpus to statistic the intent-slot co-occurrence to discover accurate correlation between intents and slots, and only in this way can an effective graph structure be determined. Besides, if there is too much noise in the training corpus, accurate graph structure cannot be built.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by the National Key Research and Development Program of China (Grant No.2021YFB3100600), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDC02040400), the Youth Innovation Promotion Association of CAS (Grant No.2021153).

References

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186.

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*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 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)*, 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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE.

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):1735–1780.

Minwoo Jeong and Gary Geunbae Lee. 2008. Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-specific dual graph neural network for multi-label text classification. 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)*, pages 3855–3864.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

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. *arXiv preprint arXiv:1909.02188*.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197. IEEE.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021b. Gl-gin: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. *arXiv preprint arXiv:2106.01925*.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. *arXiv preprint arXiv:2004.10087*.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. 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)*, pages 466–475.

Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Houfeng Wang. 2020. Graph lstm with context-gated mechanism for spoken language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9539–9546.

Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on international conference on multimedia retrieval*, pages 100–107.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

Model	MixATIS			MixSNIPS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Attention BiRNN	86.4	74.6	39.1	89.4	95.4	59.5
Slot-Gated	87.7	63.9	35.5	87.9	94.6	55.4
Bi-Model	83.9	70.3	34.4	90.7	95.6	63.4
SF-ID Network	87.4	66.2	34.9	90.6	95.0	59.9
Stack-Propagation	87.8	72.1	40.1	94.2	**96.0**	72.9
Joint Mutiple ID-SF	84.6	73.4	36.1	90.6	95.1	62.9
AGIF	86.7	74.4	40.8	94.2	95.1	74.2
GL-GIN	88.3	**76.3**	43.5	94.9	95.6	75.4
Ours	**88.5**	75.0	**48.2**	**95.0**	95.5	**75.9**

Table 3: Slot F1 score and intent accuracy results on MixATIS and MixSNIPS datasets, bold marks highest number among all models. The experimental results of all baseline models are directly cited from Qin et al. (2021b).

Model	MixATIS			MixSNIPS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Ours	88.5	75.0	48.2	95.0	95.5	75.9
w/o Intent-slot Co-occurrence	83.9	72.7	39.0	93.9	94.6	72.5
w/o Adaptive Fusion Mechanism	88.3	72.9	47.3	94.4	95.5	75.2
w/o Label-Utterance Attention	85.6	74.2	41.2	92.4	95.0	67.3
w/o Utterance Self-Attention	86.0	75.4	43.1	93.4	95.5	71.5

Table 4: Slot F1 score and intent accuracy ablation study results on MixATIS and MixSNIPS datasets.

A Additional Results

The experimental results in Table 3 show the performance of our approach compared with baselines on additional metrics for evaluating the two subtasks: F1 score of slot filling (Slot F1) and accuracy of intent detection (Intent Acc). We can see that our approach achieves comparable results compared with the best baseline on the slot F1 score and intent accuracy. It is worth noting that our approach makes significant improvements on the overall accuracy. Because the global intent-slot co-occurrence across the entire training corpus help the model enhance the joint of the two subtasks. That is to say, in the case that intents (slots) prediction is correct, our approach can better guide the prediction of slots (intents), so that our model can accurately predict all intents and slots of an utterance, hence improving the overall accuracy. Besides, Table 4 shows additional ablation study results of our approach on the slot F1 score and intent accuracy.