Co-guiding Net: Achieving Mutual Guidances between Multiple Intent Detection and Slot Filling via Heterogeneous Semantics-Label Graphs

Bowen Xing[1,2] and **Ivor W. Tsang**[2,1]

[1]Australian Artificial Intelligence Institute, University of Technology Sydney, Australia
[2]Centre for Frontier Artificial Intelligence Research, A*STAR, Singapore
bwxing714@gmail.com, ivor.tsang@gmail.com

Abstract

Recent graph-based models for joint multiple intent detection and slot filling have obtained promising results through modeling the guidance from the prediction of intents to the decoding of slot filling. However, existing methods (1) only model the *unidirectional guidance* from intent to slot; (2) adopt *homogeneous graphs* to model the interactions between the slot semantics nodes and intent label nodes, which limit the performance. In this paper, we propose a novel model termed Co-guiding Net, which implements a two-stage framework achieving the *mutual guidances* between the two tasks. In the first stage, the initial estimated labels of both tasks are produced, and then they are leveraged in the second stage to model the mutual guidances. Specifically, we propose two *heterogeneous graph attention networks* working on the proposed two *heterogeneous semantics-label graphs*, which effectively represent the relations among the semantics nodes and label nodes. Experiment results show that our model outperforms existing models by a large margin, obtaining a relative improvement of 19.3% over the previous best model on Mix-ATIS dataset in overall accuracy.

1 Introduction

Spoken language understanding (SLU) (Young et al., 2013) is a fundamental task in dialog systems. Its objective is to capture the comprehensive semantics of user utterances, and it typically includes two subtasks: intent detection and slot filling (Tur and De Mori, 2011). Intent detection aims to predict the intention of the user utterance and slot filling aims to extract additional information or constraints expressed in the utterance.

Recently, researchers discovered that these two tasks are closely tied, and a bunch of models (Goo et al., 2018; Li et al., 2018; Liu et al., 2019a; E et al., 2019; Qin et al., 2019) are proposed to combine the single-intent detection and slot filling in multi-task frameworks to leverage their correlations.
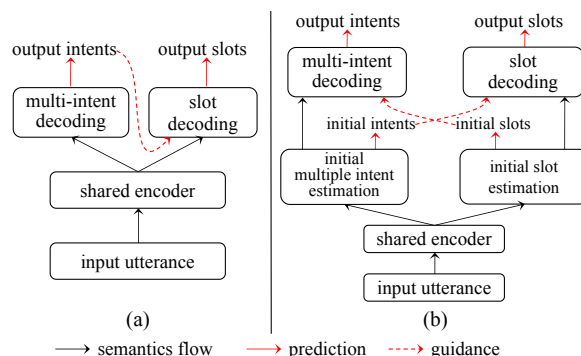

Figure 1: (a) Previous framework which only models the unidirectional guidance from multi-intent predictions to slot filling. (b) Our framework which models the mutual guidances between the two tasks.

However, in real-world scenarios, a user usually expresses multiple intents in a single utterance. To this end, (Kim et al., 2017) begin to tackle the multi-intent detection task and (Gangadharaiah and Narayanaswamy, 2019) make the first attempt to jointly model the multiple intent detection and slot filling in a multi-task framework. (Qin et al., 2020) propose an AGIF model to adaptively integrate the fine-grained multi-intent prediction information into the autoregressive decoding process of slot filling via graph attention network (GAT) (Velickovic et al., 2018). And (Qin et al., 2021b) further propose a non-autoregressive GAT-based model which enhances the interactions between the predicted multiple intents and the slot hidden states, obtaining state-of-the-art results and significant speedup.

Despite the promising progress that existing multi-intent SLU joint models have achieved, we discover that they suffer from two main issues:

(1) **Ignoring the guidance from slot to intent**. Since previous researchers realized that "slot labels could depend on the intent" (Gangadharaiah and Narayanaswamy, 2019), existing models leverage the information of the predicted intents to guide slot filling, as shown in Fig. 1(a). However, they ig-
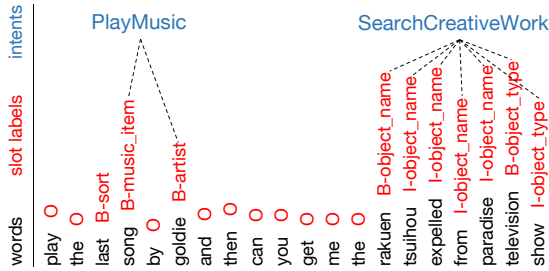
159

Figure 2: Illustration of the bidirectional interrelations between intent (blue) and slot (red) labels. The sample is retrieved from MixSNIPS dataset.

nore that slot labels can also guide the multi-intent detection task. Based on our observations, multi-intent detection and slot filling are bidirectionally interrelated and can mutually guide each other. For example, in Fig 2, not only the intents can indicate the slots, but also the slots can infer the intents. However, in previous works, the only guidance that the multiple intent detection task can get from the joint model is sharing the basic semantics with the slot filling task. As a result, the lack of guidance from slot to intent limits multiple intent detection, and so the joint task.

(2) **Node and edge ambiguity in the semantics-label graph.** (Qin et al., 2020, 2021b) apply GATs over the constructed graphs to model the interactions among the slot semantics nodes and intent label nodes. However, their graphs are homogeneous, in which all nodes and edges are treated as the same type. For a slot semantics node, the information from intent label nodes and other slot semantics nodes play different roles, while the homogeneous graph cannot discriminate their specific contributions, causing ambiguity. Therefore, the heterogeneous graphs should be designed to represent the relations among the semantic nodes and label nodes to facilitate better interactions.

In this paper, we propose a novel model termed Co-guiding Net to tackle the above two issues. For the first issue, Co-guiding Net implements a two-stage framework as shown in Fig. 1 (b). The first stage produces the initial estimated labels for the two tasks and the second stage leverages the estimated labels as prior label information to allow the two tasks mutually guide each other. For the second issue, we propose two heterogeneous semantics-label graphs (HSLGs): (1) a slot-to-intent semantics-label graph (S2I-SLG) that effectively represents the relations among the intent semantics nodes and slot label nodes; (2)

an intent-to-slot semantics-label graph (I2S-SLG) that effectively represents the relations among the slot semantics nodes and intent label nodes. Moreover, two heterogeneous graph attention networks (HGATs) are proposed to work on the two proposed graphs for modeling the guidances from slot to intent and intent to slot, respectively. Experiment results show that our Co-guiding Net significantly outperforms previous models, and model analysis further verifies the advantages of our model.

The contributions of our work are three-fold: (1) We propose Co-guiding Net[1], which implements a two-stage framework allowing multiple intent detection and slot filling mutually guide each other. We make the first attempt to achieve the mutual guidances between the two tasks. (2) We propose two heterogeneous semantics-label graphs as appropriate platforms for interactions between semantics nodes and label nodes. And we propose two heterogeneous graph attention networks to model the mutual guidances between the two tasks. (3) Experiment results demonstrate that our model achieves new state-of-the-art performance.

2 Co-guiding

Problem Definition Given a input utterance denoted as $U = \{u_i\}_1^n$, multiple intent detection can be formulated as a multi-label classification task that outputs multiple intent labels corresponding to the input utterance. And slot filling is a sequence labeling task that maps each u_i into a slot label.

Next, before diving into the details of Co-guiding Net's architecture, we first introduce the construction of the two heterogeneous graphs.

2.1 Graph Construction

2.1.1 Slot-to-Intent Semantics-Label Graph

To provide an appropriate platform for modeling the guidance from the estimated slot labels to multiple intent detection, we design a slot-to-intent semantics-label graph (S2I-SLG), which represents the relations among the semantics of multiple intent detection and the estimated slot labels. S2I-SLG is a heterogeneous graph and an example is shown in Fig. 3 (a). It contains two types of nodes: intent semantics nodes (e.g., I_1, ..., I_5) and **s**lot **l**abel (SL) nodes (e.g., SL_1, ..., SL_5). And there are four types of edges in S2I-SLG, as shown in Fig. 3 (b). Each edge type corresponds to an individual kind of information aggregation on the graph.
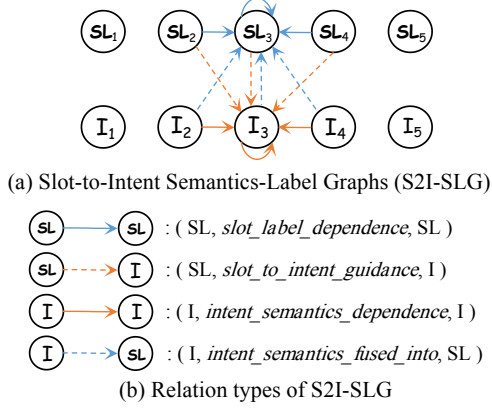
---

[1]https://github.com/XingBowen714/Co-guiding

160

(a) Slot-to-Intent Semantics-Label Graphs (S2I-SLG)

$SL \longrightarrow SL$: (SL, *slot_label_dependence*, SL)

$SL \dashrightarrow I$: (SL, *slot_to_intent_guidance*, I)

$I \longrightarrow I$: (I, *intent_semantics_dependence*, I)

$I \dashrightarrow SL$: (I, *intent_semantics_fused_into*, SL)

(b) Relation types of S2I-SLG

Figure 3: The illustration of S2I-SLG and its relation types. w.l.o.g, only the edges directed into SL_3 and I_3 are shown, and the local window size is 1.

(a) Intent-to-Slot Semantics-Label Graphs (I2S-SLG)

$IL \longrightarrow IL$: (IL, *intent_label_dependence*, IL)

$IL \dashrightarrow S$: (IL, *intent_to_slot_guidance*, S)

$S \longrightarrow S$: (S, *slot_semantics_dependence*, S)

$S \dashrightarrow IL$: (S, *slot_semantics_fused_into*, IL)

(b) Relation types of I2S-SLG

Figure 4: The illustration of I2G-SLG and its relation types. w.l.o.g, only the edges directed into IL_3 and S_3 are shown, and the local window size is 1.

Mathematically, the S2I-SLG can be denoted as $\mathcal{G}_{s2i} = (\mathcal{V}_{s2i}, \mathcal{E}_{s2i}, \mathcal{A}_{s2i}, \mathcal{R}_{s2i})$, in which $\mathcal{V}_{s2i}$ is the set of all nodes, $\mathcal{E}_{s2i}$ is the set of all edges, $\mathcal{A}_{s2i}$ is the set of two node types and $\mathcal{R}_{s2i}$ is the set of four edge types. Each node v_{s2i} and each edge e_{s2i} are associated with their type mapping functions $\tau(v_{s2i}) : \mathcal{V}_{s2i} \to \mathcal{A}_{s2i}$ and $\phi(e_{s2i}) : \mathcal{E}_{s2i} \to \mathcal{R}_{s2i}$. For instance, in Fig. 3, the SL_2 node belongs to $\mathcal{V}_{s2i}$, while its node type SL belongs to $\mathcal{A}_{s2i}$; the edge from SL_2 to I_3 belongs to $\mathcal{E}_{s2i}$, while its edge type *slot_to_intent_guidance* belongs to $\mathcal{R}_{s2i}$. Besides, edges in S2I-SLG are based on local connections. For example, node I_i is connected to $\{I_{i-w}, ..., I_{i+w}\}$ and $\{SL_{i-w}, ..., SL_{i+w}\}$, where w is a hyper-parameter of the local window size.

2.1.2 Intent-to-Slot Semantics-Label Graph

To present a platform for accommodating the guidance from the estimated intent labels to slot filling, we design an intent-to-slot semantics-label graph (I2S-SLG) that represents the relations among the slot semantics nodes and the intent label nodes. I2S-SLG is also a heterogeneous graph and an example is shown in Fig. 4 (a). It contains two types of nodes: slot semantics nodes (e.g., $S_1, ..., S_5$) and **i**ntent **l**abel (IL) nodes (e.g., $IL_1, ..., IL_5$). And Fig. 4 (b) shows the four edge types. Each edge type corresponds to an individual kind of information aggregation on the graph.

Mathematically, the I2S-SLG can be denoted as $\mathcal{G}_{i2s} = (\mathcal{V}_{i2s}, \mathcal{E}_{i2s}, \mathcal{A}_{i2s}, \mathcal{R}_{i2s})$. Each node v_{i2s} and each edge e_{i2s} are associated with their type mapping functions $\tau(v_{i2s})$ and $\phi(e_{i2s})$. The connections in I2S-SLG are a little different from S2I-SLG. Since intents are sentence-level, each IL node is globally connected with all nodes. For
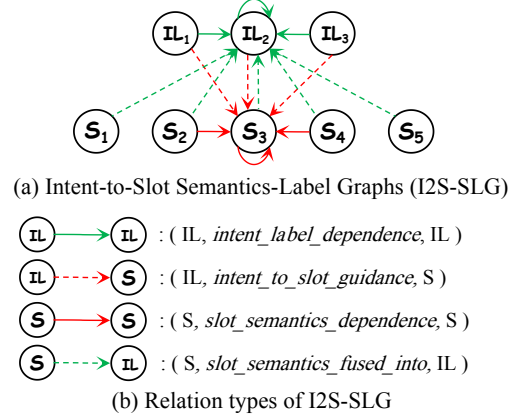
S_i node, it is connected to $\{S_{i-w}, ..., S_{i+w}\}$ and $\{IL_1, ..., IL_m\}$, where w is the local window size and m is the number of estimated intents.

2.2 Model Architecture

In this section, we introduce the details of our Co-guiding Net, whose architecture is shown in Fig.5.

2.2.1 Shared Self-Attentive Encoder

Following (Qin et al., 2020, 2021b), we adopt a shared self-attentive encoder to produce the initial hidden states containing the basic semantics. It includes a BiLSTM and a self-attention module. BiLSTM captures the temporal dependencies:

$$h_i = \text{BiLSTM}\left(x_i, h_{i-1}, h_{i+1}\right) \quad (1)$$

where x_i is the word vector of u_i. Now we obtain the context-sensitive hidden states $\hat{\boldsymbol{H}} = \{\hat{h}_i\}_1^n$.

Self-attention captures the global dependencies:

$$\boldsymbol{H}' = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V} \quad (2)$$

where $\boldsymbol{H}'$ is the global contextual hidden states output by self-attention; $\boldsymbol{Q}, \boldsymbol{K}$ and $\boldsymbol{V}$ are matrices obtained by applying different linear projections on the input utterance word vector matrix.

Then we concatenate the output of BiLSTM and self-attention to form the output of the shared self-attentive encoder: $\boldsymbol{H} = \hat{\boldsymbol{H}} \| \boldsymbol{H}'$, where $\boldsymbol{H} = \{h_i\}_1^n$ and $\|$ denotes concatenation operation.

2.2.2 Initial Estimation

Multiple Intent Detection To obtain the task-specific features for multiple intent detection, we apply a BiLSTM layer over $\boldsymbol{H}$:

$$h_i^{[I,0]} = \text{BiLSTM}_{\text{I}}\left(h_i, h_{i-1}^{[I,0]}, h_{i+1}^{[I,0]}\right) \quad (3)$$
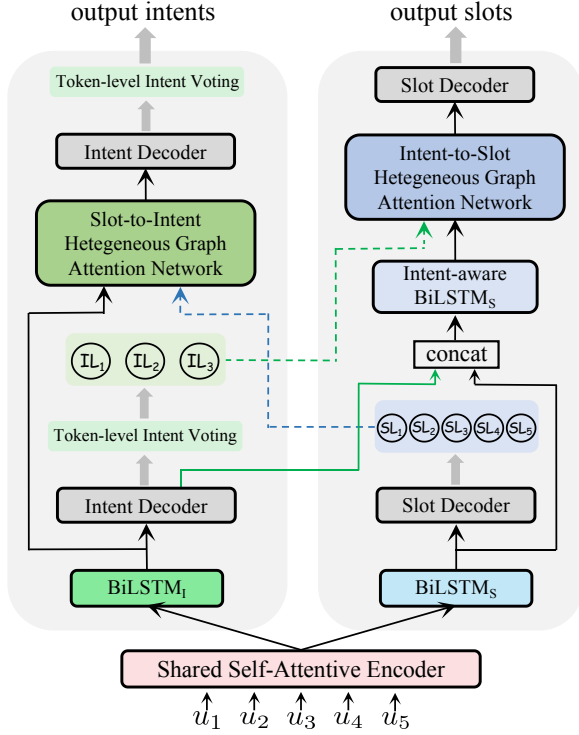
Figure 5: The architecture of Co-guiding Net. Each HGAT is triggered by its own task's semantics and the counterpart's predicted labels. The green and blue dashed arrow lines denote the projected label representations from the predicted intents and slots, respectively. The green solid arrow line denotes the intent distribution generated by the Intent Decoder at the first stage.

Following (Qin et al., 2020, 2021b), we conduct token-level multi-intent detection. Each $h_i^{[I,0]}$ is fed into the intent decoder. Specifically, the intent label distributions of the i-th word are obtained by:

$$y_i^{[I,0]} = \text{sigmoid}\left(\boldsymbol{W}_I^1\left(\sigma(\boldsymbol{W}_I^2\boldsymbol{h}_i^{[I,0]} + \boldsymbol{b}_I^2)\right) + \boldsymbol{b}_I^1\right) \quad (4)$$

where σ denotes the non-linear activation function; W_* and b_* are model parameters.

Then the estimated sentence-level intent labels $\{\text{IL}_1, ..., \text{IL}_m\}$ are obtained by the token-level intent voting (Qin et al., 2021b).

Slot Filling (Qin et al., 2021b) propose a non-autoregressive paradigm for slot filling decoding, which achieves significant speedup. In this paper, we also conduct parallel slot filling decoding.

We first apply a BiLSTM over $\boldsymbol{H}$ to obtain the task-specific features for slot filling:

$$h_i^{[S,0]} = \text{BiLSTM}_\mathbf{S}(h_i, h_{i-1}^{[S,0]}, h_{i+1}^{[S,0]}) \quad (5)$$

Then use a softmax classifier to generate the slot label distribution for each word:

$$y_i^{[S,0]} = \text{softmax}\left(\boldsymbol{W}_S^1\left(\sigma(\boldsymbol{W}_S^2\boldsymbol{h}_i^{[S,0]} + \boldsymbol{b}_S^2)\right) + \boldsymbol{b}_S^1\right) \quad (6)$$

And the estimated slot label for each word is obtained by $\text{SL}_i = \arg\max(y_i^{[S,0]})$.

2.2.3 Heterogeneous Graph Attention Network

State-of-the-art models (Qin et al., 2020, 2021b) use a homogeneous graph to connect the semantic nodes of slot filling and the intent label nodes. And GAT (Velickovic et al., 2018) is adopted to achieve information aggregation. In Sec. 1, we propose that this manner cannot effectively learn the interactions between one task's semantics and the estimated labels of the other task. To tackle this issue, we propose two heterogeneous graphs (S2I-SLG and I2S-SLG) to effectively represent the relations among the semantic nodes and label nodes. To model the interactions between semantics and labels on the proposed graphs, we propose a Heterogeneous Graph Attention Network (HGAT). When aggregating the information into a node, HGAT can discriminate the specific information from different types of nodes along different relations. And two HGATs (S2I-HGAT and I2S-HGAT) are applied on S2I-SLG and I2S-SLG, respectively. Specifically, S2I-HGAT can be formulated as follows:

$$h_i^{l+1} = \overset{K}{\underset{k=1}{\|}} \sigma\left(\sum_{j \in \mathcal{N}_{s2i}^i} W_{s2i}^{[r,k,1]}\alpha_{ij}^{[r,k]}h_j^l\right), r = \phi\left(e_{s2i}^{[j,i]}\right)$$

$$\alpha_{ij}^{[r,k]} = \frac{\exp\left(\left(W_{s2i}^{[r,k,2]}h_i^l\right)\left(W_{s2i}^{[r,k,3]}h_j^l\right)^\top/\sqrt{d}\right)}{\sum_{u \in \mathcal{N}_{s2i}^{r,i}} \exp\left(\left(W_{s2i}^{[r,k,2]}h_i^l\right)\left(W_{s2i}^{[r,k,3]}h_u^l\right)^\top/\sqrt{d}\right)} \quad (7)$$

where K denotes the total head number; $\mathcal{N}_{s2i}^i$ denotes the set of incoming neighbors of node i on S2I-SLG; $W_{s2i}^{[r,k,*]}$ are weight matrices of edge type r on the k-th head; $e_{s2i}^{[j,i]}$ denotes the edge from node j to node i on S2I-SLG; $\mathcal{N}_{s2i}^{r,i}$ denotes the nodes connected to node i with r-type edges on S2I-SLG; d is the dimension of node hidden state.

I2S-HGAT can be derived like Eq. 7.

2.2.4 Intent Decoding with Slot Guidance

In the first stage, we obtain the initial intent features $H^{[I,0]} = \{h_i^{I,0}\}_i^n$ and the initial estimated slot labels sequence $\{\text{SL}_1, ..., \text{SL}_n\}$. Now we project the slot labels into vector form using the slot label embedding matrix, obtaining $E_{sl} = \{e_{sl}^1, ..., e_{sl}^n\}$.

Then we feed $H^{[I,0]}$ and E_{sl} into S2I-HGAT to model their interactions, allowing the estimated slot label information to guide the intent decoding:

$$H^{[I,L]} = \text{S2I-HGAT}\left([H^{[I,0]}, E_{sl}], \mathcal{G}_{s2i}, \theta_I\right) \quad (8)$$

162

where $[H^{[I,0]}, E_{sl}]$ denotes the input node representation; θ_I denotes S2I-HGAT's parameters. L denotes the total layer number.

Finally, $H^{[I,L]}$ is fed to intent decoder, producing the intent label distributions for the utterance words: $Y^{[I,1]} = \{y_i^{[I,1]}, ..., y_n^{[I,1]}\}$. And the final output sentence-level intents are obtained via applying token-level intent voting over $Y^{[I,1]}$.

2.2.5 Slot Decoding with Intent Guidance

Intent-aware BiLSTM Since the B-I-O tags of slot labels have temporal dependencies, we use an intent-aware BiLSTM to model the temporal dependencies among slot hidden states with the guidance of estimated intents:

$$\tilde{h}_i^{[S,0]} = \text{BiLSTM}(y_i^{[I,0]} \| h_i^{[S,0]}, \tilde{h}_{i-1}^{[S,0]}, \tilde{h}_{i+1}^{[S,0]}) \quad (9)$$

I2S-HGAT We first project the estimated intent labels $\{\text{IL}_j\}_1^m$ into vectors using the intent label embedding matrix, obtaining $E_{il} = \{e_{il}^1, ..., e_{il}^m\}$. Then we feed $\tilde{H}^S$ and E_{il} into I2S-HGAT to model their interactions, allowing the estimated intent label information to guide the slot decoding:

$$H^{[S,L]} = \text{I2S-HGAT}\left([\tilde{H}^S, E_{il}], \mathcal{G}_{i2s}, \theta_S\right) \quad (10)$$

where $[\tilde{H}^{[S]}, E_{il}]$ denotes the input node representation; θ_S denotes I2S-HGAT's parameters.

Finally, $H^{[S,L]}$ is fed to slot decoder, producing the slot label distributions for each word: $Y^{[S,1]} = \{y_i^{[S,1]}, ..., y_n^{[S,1]}\}$. And the final output slot labels are obtained by applying $\arg\max$ over $Y^{[S,1]}$.

2.2.6 Training Objective

Loss Function The loss function for multiple intent detection is:

$$\text{CE}(\hat{y}, y) = \hat{y}\log(y) + (1-\hat{y})\log(1-y)$$
$$\mathcal{L}_I = \sum_{t=0}^{1}\sum_{i=1}^{n}\sum_{j=1}^{N_I} \text{CE}\left(\hat{y}_i^I[j], y_i^{[I,t]}[j]\right) \quad (11)$$

And the loss function for slot filling is:

$$\mathcal{L}_S = \sum_{t=0}^{1}\sum_{i=1}^{n}\sum_{j=1}^{N_S} \hat{y}_i^S[j]\log\left(y_i^{[S,t]}[j]\right) \quad (12)$$

where N_I and N_S denote the total numbers of intent labels and slot labels; $\hat{y}_i^I$ and $\hat{y}_i^S$ denote the ground-truth intent labels and slot labels.

Margin Penalty The core of our model is to let the two tasks mutually guide each other. Intuitively, the predictions in the second stage should be better than those in the first stage. To force our model obey this rule, we design a margin penalty ($\mathcal{L}^{mp}$) for each task, whose aim is to improve the probabilities of the correct labels. Specifically, the formulations of $\mathcal{L}_I^{mp}$ and $\mathcal{L}_S^{mp}$ are:

$$\mathcal{L}_I^{mp} = \sum_{i=1}^{n}\sum_{j=1}^{N_I} \hat{y}_i^I[j] \, \max\left(0, y_i^{[I,0]}[j] - y_i^{[I,1]}[j]\right)$$
$$\mathcal{L}_S^{mp} = \sum_{i=1}^{n}\sum_{j=1}^{N_S} \hat{y}_i^S[j] \max\left(0, y_i^{[S,0]}[j] - y_i^{[S,1]}[j]\right) \quad (13)$$

Model Training The training objective $\mathcal{L}$ is the weighted sum of loss functions and margin regularizations of the two tasks:

$$\mathcal{L} = \gamma\left(\mathcal{L}_I + \beta_I\mathcal{L}_I^{mp}\right) + (1-\gamma)\left(\mathcal{L}_S + \beta_S\mathcal{L}_S^{mp}\right) \quad (14)$$

where γ is the coefficient balancing the two tasks; β_I and β_S are the coefficients of the margin regularization for the two tasks.

3 Experiments

3.1 Datasets and Metrics

Following previous works, MixATIS and MixSNIPS (Hemphill et al., 1990; Coucke et al., 2018; Qin et al., 2020) are taken as testbeds. MixATIS includes 13,162 utterances for training, 756 ones for validation and 828 ones for testing. MixSNIPS includes 39,776 utterances for training, 2,198 ones for validation and 2,199 ones for testing.

As for evaluation metrics, following previous works, we adopt accuracy (Acc) for multiple intent detection, F1 score for slot filling, and overall accuracy for the sentence-level semantic frame parsing. Overall accuracy denotes the ratio of sentences whose intents and slots are all correctly predicted.

3.2 Implementation Details

Following previous works, the word and label embeddings are trained from scratch[2]. The dimensions of word embedding, label embedding, and hidden state are 256 on MixATIS, while on MixSNIPS they are 256, 128, and 256. The layer number of all GNNs is 2. Adam (Kingma and Ba, 2015) is used to train our model with a learning rate of $1e^{-3}$ and a weight decay of $1e^{-6}$. As for the coefficients Eq.14, γ is 0.9 on MixATIS and 0.8 on

---

[2]Due to space limitation, the experiments using pre-trained language model as the encoder are presented in Appendix.

Models	MixATIS			MixSNIPS		
	Overall(Acc)	Slot (F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
Attention BiRNN (Liu and Lane, 2016)	39.1	86.4	74.6	59.5	89.4	95.4
Slot-Gated (Goo et al., 2018)	35.5	87.7	63.9	55.4	87.9	94.6
Bi-Model (Wang et al., 2018)	34.4	83.9	70.3	63.4	90.7	95.6
SF-ID (E et al., 2019)	34.9	87.4	66.2	59.9	90.6	95.0
Stack-Propagation (Qin et al., 2019)	40.1	87.8	72.1	72.9	94.2	96.0
Joint Multiple ID-SF (Gangadharaiah and Narayanaswamy, 2019)	36.1	84.6	73.4	62.9	90.6	95.1
AGIF (Qin et al., 2020)	40.8	86.7	74.4	74.2	94.2	95.1
GL-GIN (Qin et al., 2021b)	43.0	88.2	76.3	73.7	94.0	95.7
Co-guiding Net (ours)	51.3†	89.8†	79.1†	77.5†	95.1†	97.7†

Table 1: Results comparison. † denotes our model significantly outperforms baselines with $p < 0.01$ under t-test.

MixSNIPS; on both datasets, β_I is $1e^{-6}$ and β_S is $1e^0$. The model performing best on the dev set is selected then we report its results on the test set. All experiments are conducted on RTX 6000. Our source code will be released.

3.3 Main Results

The performance comparison of Co-guiding Net and baselines are shown in Table 1, from which we have the following observations:

(1) Co-guiding Net gains significant and consistent improvements on all tasks and datasets. Specifically, on MixATIS dataset, it overpasses the previous state-of-the-art model GL-GIN by 19.3%, 1.8%, and 3.7% on sentence-level semantic frame parsing, slot filling, and multiple intent detection, respectively; on MixSNIPS dataset, it overpasses GL-GIN by 5.2%, 1.2% and 2.1% on sentence-level semantic frame parsing, slot filling and multiple intent detection, respectively. This is because our model achieves the mutual guidances between multiple intent detection and slot filling, allowing the two tasks to provide crucial clues for each other. Besides, our designed HSLGs and HGATs can effectively model the interactions among the semantics nodes and label nodes, extracting the indicative clues from initial predictions.

(2) Co-guiding Net achieves a larger improvement on multiple intent detection than slot filling. The reason is that except for the guidance from multiple intent detection to slot filling, our model also achieves the guidance from slot filling to multiple intent detection, while previous models all ignore this. Besides, previous methods model the semantics-label interactions by homogeneous graph and GAT, limiting the performance. Differently, our model uses the heterogeneous semantics-label graphs to represent different relations among the semantic nodes and the label

nodes, then applies the proposed HGATs over the graphs to achieve the interactions. Consequently, their performances (especially on multiple intent detection) are significantly inferior to our model.

(3) The improvements in overall accuracy are much sharper. We suppose the reason is that the achieved mutual guidances make the two tasks deeply coupled and allow them to stimulate each other using their initial predictions. For each task, its final outputs are guided by its and another task's initial predictions. By this means, the correct predictions of the two tasks can be better aligned. As a result, more test samples get correct sentence-level semantic frame parsing results, and then overall accuracy is boosted.

3.4 Model Analysis

We conduct a set of ablation experiments to verify the advantages of our work from different perspectives, and the results are shown in Table 2.

3.4.1 Effect of Slot-to-Intent Guidance

One of the core contributions of our work is achieving the mutual guidances between multiple intent detection and slot filling, while previous works only leverage the one-way message from intent to slot. Therefore, compared with previous works, one of the advantages of our work is modeling the slot-to-intent guidance. To verify this, we design a variant termed *w/o S2I-guidance* and its result is shown in Table 2. We can observe that Intent Acc drops by 2.0% on MixATIS and 0.8% on MixSNIPS. Moreover, Overall Acc drops more significantly: 3.6% on MixATIS and 0.9% on MixSNIPS. This proves that the guidance from slot to intent can effectively benefit multiple intent detection, and achieving the mutual guidances between the two tasks can significantly improve Overall Acc.

Besides, although both of *w/o S2I-guidance* and

Models	MixATIS			MixSNIPS		
	Overall(Acc)	Slot (F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
Co-guiding Net	**51.3**	**89.8**	**79.1**	**77.5**	**95.1**	**97.7**
w/o S2I-guidance	47.7 (↓3.6)	88.8 (↓1.0)	77.1 (↓2.0)	76.6 (↓0.9)	94.7 (↓0.4)	96.9 (↓0.8)
w/o I2S-guidance	47.7 (↓3.6)	88.7 (↓1.1)	77.5 (↓1.6)	76.5 (↓1.0)	94.9 (↓0.2)	97.5 (↓0.2)
w/o relations	46.0 (↓5.3)	88.3 (↓1.5)	77.8 (↓1.3)	76.3 (↓1.2)	94.7 (↓0.5)	97.2 (↓0.4)
+ Local Slot-aware GAT	51.1 (↓0.2)	89.4 (↓0.4)	79.0 (↓0.1)	75.9 (↓1.6)	94.7 (↓0.4)	96.4 (↓1.4)

Table 2: Results of ablation experiments.

GL-GIN only leverage the one-way message from intent to slot, *w/o S2I-guidance* outperforms GL-GIN by large margins. We attribute this to our proposed heterogeneous semantics-label graphs and heterogeneous graph attention networks, whose advantages are verified in Sec. 3.4.3.

3.4.2 Effect of Intent-to-Slot Guidance

To verify the effectiveness of intent-to-slot guidance, we design a variant termed *w/o I2S-guidance* and its result is shown in Table 2. We can find that the intent-to-slot guidance has a significant impact on performance. Specifically, *w/o I2S-guidance* cause nearly the same extent of performance drop on Overall Acc, proving that both of the intent-to-slot guidance and slot-to-intent guidance are indispensable and achieving the mutual guidances can significantly boost the performance.

3.4.3 Effect of HSLGs and HGATs

In this paper, we design two HSLGs: (i.e., S2I-SLG, I2S-SLG) and two HGATs (i.e., S2I-HGAT, I2S-HGAT). To verify their effectiveness, we design a variant termed *w/o relations* by removing the relations on the two HSLGs. In this case, S2I-SLG/I2S-SLG collapses to a homogeneous graph, and S2I-HGAT/I2S-HGAT collapses to a general GAT based on multi-head attentions. From Table 2, we can observe that *w/o relations* obtains dramatic drops on all metrics on both datasets. The apparent performance gap between *w/o relations* and Co-guiding Net verifies that (1) our proposed HSLGs can effectively represent the different relations among the semantics nodes and label nodes, providing appropriate platforms for modeling the mutual guidances between the two tasks; (2) our proposed HGATs can sufficiently and effectively model interactions between the semantics and indicative label information via achieving the relation-specific attentive information aggregation on the HSLGs.

Besides, although *w/o relations* obviously un-

derperforms Co-guiding Net, it still significantly outperforms all baselines. We attribute this to the fact that our model achieves the mutual guidances between the two tasks, which allows them to promote each other via cross-task correlations.

3.4.4 Effect of I2S-HGAT for Capturing Local Slot Dependencies

Qin et al. (2021b) propose a Local Slot-aware GAT module to alleviate the uncoordinated slot problem (e.g., *B-singer* followed by *I-song*) (Wu et al., 2020) caused by the non-autoregressive fashion of slot filling. And the ablation study in (Qin et al., 2021b) proves that this module effectively improves the slot filling performance by modeling the local dependencies among slot hidden states. In their model (GL-GIN), the local dependencies are modeled in both of the local slot-aware GAT and subsequent global intent-slot GAT. We suppose the reason why GL-GIN needs the local Slot-aware GAT is that the global intent-slot GAT in GL-GIN cannot effectively capture the local slot dependencies. GL-GIN's global slot-intent graph is homogeneous, and the GAT working on it treats the slot semantics nods and the intent label nodes equally without discrimination. Therefore, each slot hidden state receives indiscriminate information from both of its local slot hidden states and all intent labels, making it confusing to capture the local slot dependencies. In contrast, we believe our I2S-HLG and I2S-HGAT can effectively capture the slot local dependencies along the specific *slot_semantics_dependencies* relation, which is modeled together with other relations. Therefore, our Co-guiding Net does not include another module to capture the slot local dependencies.

To verify this, we design a variant termed *+Local Slot-aware GAT*, which is implemented by augmenting Co-guiding Net with the Local Slot-aware GAT (Qin et al., 2021b) located after the Intent-aware BiLSTM$_s$ (the same position with GL-GIN). And its result is shown in Table 2. We can observe

Utterance: New York city to Las Vegas and Memphis to Las Vegas on Sunday

(A)

Stage I
Slot: B-fromloc.city_name I-fromloc.city_name I-fromloc.city_name O B-toloc.city_name I-toloc.city_name O B-fromloc.city_name O B-toloc.city_name I-toloc.city_name O B-depart_date.day_name
Intent: atis_airport atis_flight
Sentence-level Semantic Frame Parsing: Incorrect

Stage II
Slot: B-fromloc.city_name I-fromloc.city_name I-fromloc.city_name O B-toloc.city_name I-toloc.city_name O B-fromloc.city_name O B-toloc.city_name I-toloc.city_name O B-depart_date.day_name
Intent: atis_flight
Sentence-level Semantic Frame Parsing: Correct

Utterance: Tell me about the m80 aircraft and also how much is the limousine service in Boston

(B)

Stage I
Slot: O O O O B-mod O O O O O O B-transport_type O O B-city_name
Intent: atis_aircraft atis_ground_fare
Sentence-level Semantic Frame Parsing: Incorrect

Stage II
Slot: O O O O B-aircraft_code O O O O O O B-transport_type O O B-city_name
Intent: atis_aircraft atis_ground_fare
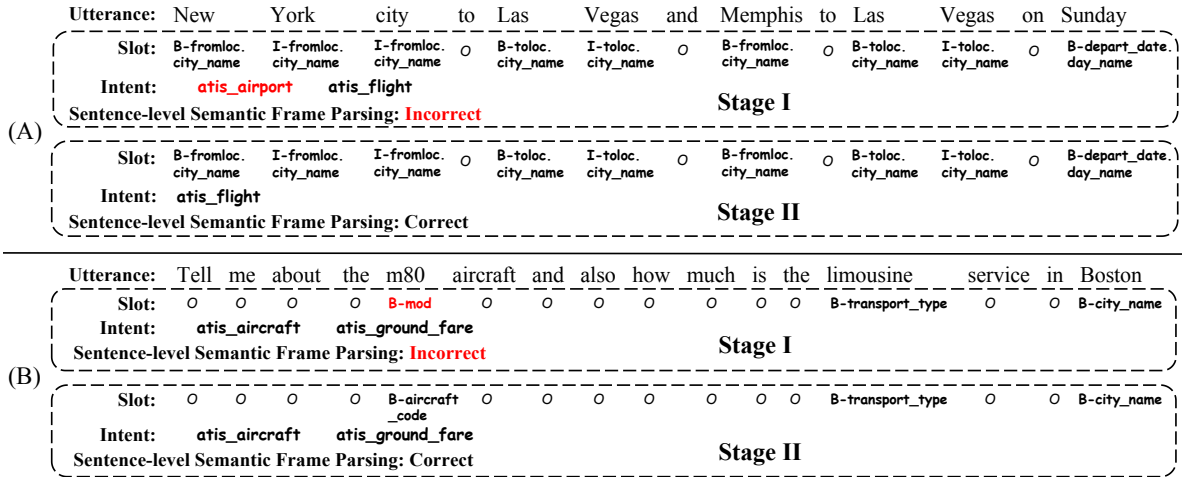Sentence-level Semantic Frame Parsing: Correct

Figure 6: Case study of slot-to-intent guidance (A) and intent-to-slot guidance (B). Red color denotes error.

that not only the Local Slot-aware GAT does not bring improvement, it even causes performance drops. This proves that our I2S-HGAT can effectively capture the local slot dependencies.

3.5 Case Study

To demonstrate how our model allows the two tasks to guide each other, we present two cases in Fig. 6.

Slot-to-Intent Guidance From Fig. 6 (A), we can observe that in the first stage, all slots are correctly predicted, while multiple intent detection obtains a redundant intent atis_airport. In the second stage, our proposed S2I-HGAT operates on S2I-HLG. It aggregates and analyzes the slot label information from the slot predictions of the first stage, extracting the indicative information that most slot labels are about city_name while no information about *airport* is mentioned. Then this beneficial guidance information is passed into intent semantics nodes whose representations are then fed to the intent decoder for prediction. In this way, the guidance from slot filling helps multiple intent detection predict correctly.

Intent-to-Slot Guidance In the example shown in Fig. 6 (B), in the first stage, correct intents are predicted, while there is an error in the predicted slots. In the second stage, our proposed I2S-HGAT operates on I2S-HLG. It comprehensively analyzes the indicative information of *airecraft* from both of slot semantics node *aircraft* and intent label node atis_aircraft. Then this beneficial guidance information is passed into the slot semantics of *m80*, whose slot is therefore correctly inferred.

4 Related Work

The correlations between intent detection and slot filling have been widely recognized. To leverage them, a group of models (Zhang and Wang, 2016; Hakkani-Tür et al., 2016; Goo et al., 2018; Li et al., 2018; E et al., 2019; Liu et al., 2019a; Qin et al., 2019; Zhang et al., 2019; Wu et al., 2020; Qin et al., 2021a; Ni et al., 2021) are proposed to tackle the joint task of intent detection and slot filling in a multi-task manner. However, the intent detection modules in the above models can only handle the utterances expressing a single intent, which may not be practical in real-world scenarios, where there are usually multi-intent utterances.

To this end, Kim et al. (2017) propose a multi-intent SLU model, and (Gangadharaiah and Narayanaswamy, 2019) propose the first model to jointly model the tasks of multiple intent detection and slot filling via a slot-gate mechanism. Furthermore, as graph neural networks have been widely utilized in various tasks (Cao et al., 2019; Wang et al., 2020; Shi et al., 2021; Xing and Tsang, 2022e,a,b), they have been leveraged to model the correlations between intent and slot. Qin et al. (2020) propose an adaptive graph-interactive framework to introduce the fine-grained multiple intent information into slot filling achieved by GATs. More recently, Qin et al. (2021b) propose another GAT-based model, which includes a non-autoregressive slot decoder conducting parallel decoding for slot filling and achieves the state-of-the-art performance.

Our work also tackles the joint task of multiple intent detection and slot filling. Existing methods

only model the one-way guidance from multiple intent detection to slot filling. Besides, they adopt homogeneous graphs and vanilla GATs to achieve the interactions between the predicted intents and slot semantics. In contrast, we (1) achieve the mutual guidances between the two tasks; (2) propose the heterogeneous semantics-label graphs to represent the dependencies among the semantics and predicted labels; (3) we propose the Heterogeneous Graph Attention Network to model the semantics-label interactions on the heterogeneous semantics-label graphs.

5 Conclusion

In this paper, we propose a novel Co-guiding Net based on a two-stage framework that allows the two tasks to guide each other in the second stage using the predicted labels at the first stage. To represent the relations among the semantics node and label nodes, we propose two heterogeneous semantics-label graphs, and two heterogeneous graph attention networks are proposed to model the mutual guidances between intents and slots. Experiment results on benchmark datasets show that our model significantly outperforms previous models. Future work will focus on leveraging syntactic information to enhance utterance understanding.

Limitations

Although our Co-guiding Net achieves significant improvement over existing models, we suppose that its sentence understanding module (the self-attentive encoder) is not sufficient enough and limits the performance to some extent. In recent years, the syntactic information extracted from the sentence's syntax/dependency tree, which is output by an off-the-shelf dependency parser, has been widely leveraged to assist the models in comprehensively understanding the sentence (Wang et al., 2020; Jiang and Cohn, 2021; Tian et al., 2021; Xing and Tsang, 2022c,d). And in SLU, each utterance is a sentence, and we believe that leveraging the syntactic information can improve the performance by enriching the word representations.

Acknowledgements

References

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1452–1461, Florence, Italy. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 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*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech 2016*, pages 715–719.

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*.

Fan Jiang and Trevor Cohn. 2021. Incorporating syntax and semantics in coreference resolution with heterogeneous graph attention network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1584–1591, Online. Association for Computational Linguistics.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Proc. Interspeech 2016*, pages 685–689.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 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)*, pages 1051–1060, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Adiga, and Erik Cambria. 2021. Recent advances in deep learning based dialogue systems: A systematic survey. *arXiv preprint arXiv:2105.04387*.

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 *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 2078–2087, Hong Kong, China. Association for Computational Linguistics.

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.

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. 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 178–188, Online. Association for Computational Linguistics.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 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*, pages 1807–1816, Online. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven relation extraction with attentive graph convolutional networks. 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 4458–4471, Online. Association for Computational Linguistics.

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

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz,

Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Di Wu, Liang Ding, Fan Lu, and Jian Xie. 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)*, pages 1932–1937, Online. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022a. DARER: Dual-task temporal relational recurrent reasoning network for joint dialog sentiment classification and act recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3611–3621, Dublin, Ireland. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022b. Dignet: Digging clues from local-global interactive graph for aspect-level sentiment classification. *arXiv preprint arXiv:2201.00989*.

Bowen Xing and Ivor Tsang. 2022c. Neural subgraph explorer: Reducing noisy information via target-oriented syntax graph pruning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4425–4431.

Bowen Xing and Ivor W Tsang. 2022d. Out of context: A new clue for context modeling of aspect-based sentiment analysis. *Journal of Artificial Intelligence Research*, 74:627–659.

Bowen Xing and Ivor W. Tsang. 2022e. Understand me, if you refer to aspect knowledge: Knowledge-aware gated recurrent memory network. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1092–1102.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2993–2999. IJCAI/AAAI Press.

A Experiments using Pre-trained Language Model

A.1 Settings

To evaluate Co-guiding Net's performance based on the pre-trained language model, we use the pre-trained RoBERTa (Liu et al., 2019b) encoder to replace the original self-attentive encoder. We adopt the pre-trained RoBERTa-base version provided by Transformers (Wolf et al., 2020). For each word, its first subwords' hidden state generated by RoBERTa is taken as the word representation. AdamW (Loshchilov and Hutter, 2019) optimizer is used for model training with the default setting, and RoBERTa is fine-tuned with model training. Other model components are identical to the Co-guiding Net based on LSTM, and we use the same hyper-parameters of the model rather than search for the optimal ones for RoBERTa+Co-guiding Net due to our limited computation resource.

A.2 Results

Table 3 shows the result comparison of Co-guiding Net, RoBERTa+Co-guiding Net, and their state-of-the-art counterparts: AGIF, GL-GIN, RoBERTa+AGIF, and RoBERTa+GL-GIN. We can find that although RoBERTa boosts the models' performance, RoBERTa+Co-guiding Net still significantly outperforms RoBERTa+AGIF and RoBERTa+GL-GIN. This can be attributed to the fact that although the pre-trained language model (PTLM) can enhance the word representations, it cannot achieve the guidance between the two tasks or the interactions between the semantics and label information, which are exactly the advantages of our Co-guiding Net. Therefore, collaborating with PTLM that has strong ability of language modeling, RoBERTa+Co-guiding Net gets its performance further boosted, achieving new state-of-the-art.

Models	MixATIS	MixSNIPS
	Overall(Acc)	Overall(Acc)
AGIF	40.8	74.2
GL-GIN	43.0	73.7
Co-guiding Net (ours)	51.3†	77.5†
RoBERTa+AGIF	50.0	80.7
RoBERTa+GL-GIN	53.6	82.6
RoBERTa+Co-guiding Net (ours)	57.5†	85.3†

Table 3: Results comparison. † denotes our model significantly outperforms the corresponding counterparts with $p < 0.01$ under t-test.