Focus on Interaction: A Novel Dynamic Graph Model for Joint Multiple Intent Detection and Slot Filling

Zeyuan Ding , **Zhihao Yang*** , **Hongfei Lin** and **Jian Wang**
Dalian University of Technology, Dalian, China
zeyuanding@mail.dlut.edu.cn, {yangzh, hflin, wangjian}@dlut.edu.cn

Abstract

Intent detection and slot filling are two main tasks for building a spoken language understanding (SLU) system. Since the two tasks are closely related, the joint models for the two tasks always outperform the pipeline models in SLU. However, most joint models directly incorporate multiple intent information for each token, which introduces intent noise into the sentence semantics, causing a decrease in the performance of the joint model. In this paper, we propose a Dynamic Graph Model (DGM) for joint multiple intent detection and slot filling, in which we adopt a sentence-level intent-slot interactive graph to model the correlation between the intents and slot. Besides, we design a novel method of constructing the graph, which can dynamically update the interactive graph and further alleviate the error propagation. Experimental results on several multi-intent and single-intent datasets show that our model not only achieves the state-of-the-art (SOTA) performance but also boosts the speed by three to six times over the SOTA model.

1 Introduction

Spoken language understanding (SLU) is a critical component in task-oriented dialogue systems. It aims to form a semantic frame that captures the semantics of user utterances or queries. SLU consists of two typical subtasks, intent detection and slot filling [Tur and Mori, 2011]. Intent detection captures the intention of the user and slot filling extracts additional information or constraints provided by the users. As shown in Figure 1, taking the utterance "*Book a brasserie for me*" for example, the intent detection can be formulated as a classification task to classify the intent label while the slot filling as a sequence labeling task to predict the slot label sequence.

Considering that pipeline approaches usually suffer from error propagation, the joint model has been proposed to improve sentence-level semantics via mutual enhancement between two tasks [Guo *et al.*, 2014; Hakkani-Tür *et al.*, 2016].

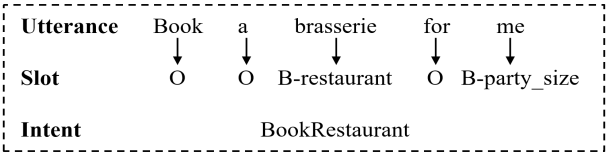

Figure 1: An example with intent and slot annotation (BIO format), which indicates the slot of restaurant and party size from an utterance with an intent BookRestaurant.

Existing researches on joint method can be mainly classified into two classes. One makes use of sharing parameters to model the relationship between intent and slot [Bing and Lane, 2016; Zhang and Wang, 2016] and other explicitly leverages intent information to guide slot filling task [Goo *et al.*, 2018; Li *et al.*, 2018; Qin *et al.*, 2019].

Despite their success, existing joint models only focus on single intent scenarios, i.e. these models are trained based on the assumption that each utterance only has one single intent. However, in real-world scenarios, users usually express multiple intents in an utterance. For example, given a user's utterance, "*Book a brasserie for me at four pm and what is the weather in neighboring*", the user expresses his intentions (*BookRestaurant* and *GetWeather*) in one utterance. Unlike the prior SLU models, Gangadharaiah and Narayanaswamy [2019] adopted a multi-task framework with the slot-gated mechanism (Joint Multiple ID-SF) for multiple intent detection and slot filling. Their model uses an intent context vector to incorporate the information of multiple intents, where each token is guided with the same complex intents information.

Although Joint Multiple ID-SF proposes an effective strategy to incorporate multiple intent information, it always introduces intent noise into the sentence semantics. To alleviate the noise issue, Qin et al. [2020] proposed an adaptive graph-interactive framework for multiple intent detection and slot filling, which builds token-level interactive graph for each token in the utterance by using all predicted intents. However, different tokens appearing in the utterance have various importance for representing the intent. For instance, in the utterance "*Book a brasserie for me at four pm*" , "*me*" is not important to the judgment of the intention(*BookRestaurant*). Their model incorporates multiple intent information to all tokens including those without contribution to intent repre-

*Corresponding author

sentation, which introduces noise into the sentence semantics and decreases in the performance of the model to some extent.

In this paper, we propose a dynamic graph network that focuses on the interactive information of intent and slot to address above two issues. The core module is the dynamic graph module, which consists of a generation graph layer and a graph attention network. The generation graph layer automatically constructs sentence-level intent-slot interactive graph by connecting the tokens with the relevant intents. To alleviate introduction of noise, the graph only incorporates intent information for tokens that play an important role in judging sentence intents. Besides, we design a novel method of constructing the graph, which can dynamically update the interactive graph until the tokens in the utterance match the correct intents. To encode multiple intents information, we introduce the graph attention network (GAT) to model the interactive graph. GAT leverages masked self-attention layers to assign different importance to neighbouring nodes. It makes a part of tokens incorporated with relevant intent information and provides prior information for slot filling. In contrast to prior work which directly incorporate multiple intents information statically, our model captures relevant intent information to construct intent-slot interactive graph. Furthermore, compared with token-level interactive graph, the sentence-level graph can achieve a 3-6 times speed gain.

We conducted the experiments on two multi-intent datasets MixATIS [Hemphill *et al.*, 1990] and MixSNIPS [Coucke *et al.*, 2018]and results indicate the effectiveness of our method. Moreover, to verify the generalization of our model, we also constructed the experiments on two single-intent datasets ATIS [Tur and Mori, 2011] and SNIPS [Coucke *et al.*, 2018]. The results show our method significantly outperforms a series of joint intent and slot methods and further verify its generalization.

To summarize, the contributions of this work are as follows:

1. We propose a dynamic graph model in SLU task, which can establish the directional interrelated connections for the two tasks and achieve the best results on several multi-intent and single-intent datasets.
2. To alleviate the noise issue, we employ sentence-level intent-slot interactive graph. The graph incorporates intent information only for tokens that play an important role in judging sentence intents.
3. Different from using external NLP tools to construct graph, we propose a novel method of graph construction, which dynamically updates the interaction graph. This method achieves a 3-6x speed up over the SOTA model.

To facilitate future research in this area, the codes are publicly available at https://github.com/dzy1011/DGM.

2 Related Work

Traditional pipeline approaches manage the two tasks separately. The intent detection can be treated as an utterance classification problem. Sarikaya et al. [2011] adopted recurrent neural networks (RNN) to solve it. Xia et al. [2018] used a capsule with self-attention for intent detection. Slot filling is formulated as a sequence labeling task, and the popular approaches are conditional random fields (CRF) [Raymond and Riccardi, 2007] and RNN [Xu and Sarikaya, 2013]. Recently, Zhong et al. [2018] introduced the self-attention mechanism for CRF-free sequential labeling. However, the above pipeline models always suffer from error propagation problem.

In consideration of the high correlation between intent detection and slot filling, the tendency is to develop a joint model [Guo *et al.*, 2014; Hakkani-Tür *et al.*, 2016] for intent detection and slot filling tasks. Existing methods make use of sharing parameters [Bing and Lane, 2016; Zhang and Wang, 2016] and applying intent information to guide the slot filling [Goo *et al.*, 2018; Li *et al.*, 2018; Qin *et al.*, 2019]. However, the above joint models mainly focus on the single intent scenario, which can not handle the complex multiple intent scenario. Gangadharaiah and Narayanaswamy [2019] first adopted a multi-task framework with the slot-gated mechanism for multiple intent detection and slot filling. To better incorporate intent information, Qin et al. [2020] proposed a token-level graph-interactive framework for multiple intent detection and slot filling.

Compared with other work, the main differences are as following: 1) We propose the sentence-level intent-slot interactive graph, which only incorporate intent information for tokens that play an important role in judging sentence intents. 2) We design a novel method of constructing the graph, which can dynamically update the interactive graph to further alleviates the error propagation. 3) Our model enhances the directional interrelated connections for the two tasks and help them promote each other mutually.

3 Method

In this section, we will describe our dynamic graph model for SLU task. The architecture of our model is demonstrated in Figure 2, which consists of a shared encoder, dynamic graph module and two decoders. First, the shared encoder (Section 3.1) encodes an utterance to obtain the shared information between intent detection and slot filling. Then, the intent detection decoder (Section 3.2) performs intent detection. Furthermore, the dynamic graph module (Section 3.3) leverages intent matrix and the intent output from the intent decoder to construct intent-slot interactive graph, and uses GAT to encode the graph. Finally, we use slot filling decoder to predict the slot. Both intent detection and slot filling are optimized simultaneously via a joint learning scheme.

3.1 Shared Encoder

Following [Qin *et al.*, 2020], our shared encoder consists of BiLSTM, which leverages temporal features within word orders, followed by a self-attention mechanism to capture the contextual information of the sequence.

BiLSTM

To capture contextual information, a bidirectional LSTM [Hochreiter and Schmidhuber, 1997] is applied to encode the utterance $X = [x_1, x_2, ..., x_n]$ (n is the number of token in the input utterance). By concatenating the forward and

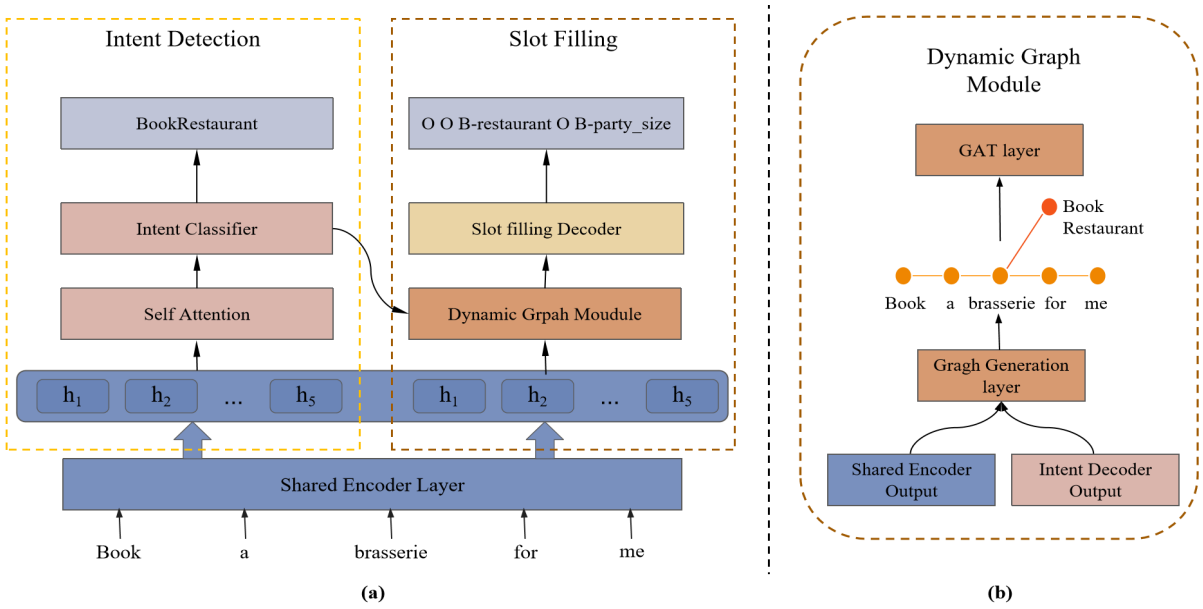

Figure 2: Illustration of our dynamic graph model for joint multiple intent detection and slot filling. It consists of one shared encoder, dynamic graph module and two decoders (a). (b) is the dynamic graph module. It leverages the intent output from the intent decoder to construct intent-slot interactive graph, and uses GAT to encode the interactive graph for slot prediction.

backward LSTM hidden states, we obtain the contextual representation $H = [h_1, h_2, ..., h_n]$.

Self-Attention

Self-attention [Vaswani *et al.*, 2017] is an effective method to capture the contextual information. In this case, we employ self-attention over word embedding to leverage context-aware features. We first map the matrix of input vectors $X \in \mathbf{R}^{n \times d}$ (d represents the mapped dimension) to queries Q, keys K and values V matrices by using different linear projections. Then the attention is used to output representation $S \in \mathbf{R}^{n \times d}$ based on the following equation:

$$S = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

For each utterance, we concatenate these two representations as the final encoding representation:

$$E = [H \parallel S] \tag{2}$$

where $E \in \mathbf{R}^{n \times 2d}$ and $\parallel$ denotes concatenation operation.

3.2 Intent Decoder

To better apply multi-intent scenarios, we treat intent detection as the multi-label classification problem. Following [Goo *et al.*, 2018], self-attention layer is applied to $E = [e_1, e_2, ..., e_n]$ to capture the contextual information for each token.

$$p_t = softmax(w_e e_t + b) \tag{3}$$

$$c = \sum_t p_t e_t \tag{4}$$

where $w_e \in \mathbf{R}^{1 \times 2d}$ is trainable parameters. c is the weighted sum of each element e_t and utilized for intent detection:

$$O^I = \sigma(W_i(LeakyReLU(W_c c) + b_c) + b_i) \tag{5}$$

where W_i, W_c are trainable parameters of the intent decoder, σ denotes the nonlinearity activation function. $O^I = [o_1^I, o_2^I, ..., o_m^I]$ is the intent output of the utterance and m represents the number of intent labels.

3.3 Dynamic Graph Module

To better incorporate intent information, we design a novel generation graph layer, which automatically extracts relevant information to construct sentence-level interactive graphs.

Generation Graph Layer

Considering that tokens may contribute differently to judging the intent of the utterance, we adopt a multiplicative attention mechanism for computing relevance between intent and token. As shown in Figure 2 (b), we embed all the intent types (e.g,*GetWeather*, *BookRestaurant*) in the same space as the word embeddings. We denote the intent that output of intent decoder as $I = [i_1, i_2, ..., i_p]$, $I \in \mathbf{R}^{p \times d_I}$, where p is the number of output intent, d_I is the dimension of the intent embeddings. We express the computation as follows:

$$t_{ij} = e_i w_I i_j \tag{6}$$

$$a_{ij} = \frac{exp(t_{ij})}{\sum_{k=0}^{i} exp(t_{kj})} \tag{7}$$

where w_I is the trainable parameters, a_{ij} is the relevance score between i^{th} intent and e_j.

Innovatively, we use hyperparameter α to measure the relevance between intent and token. If $a_{ij} > \alpha$, it means that

the token play an important role in judging the intent of the utterance. In this case, this token is directly connected to the relevant intent. In training process, we allow a token to match multiple intents, and an utterance to match multiple intents. As is shown in Figure 2 (b), "*brasserie*" and "*BookRestaurant*" have the high relevant score. Therefore, we directly connect "*brasserie*" and "*BookRestaurant*" to build an intent-slot interactive graph. During training, the interactive graph is dynamically updated according to the relevance score, leading each token to match with the correct intents. The intent-slot interactive graph contains intent information to help the subsequent slot filling task.

To represent the edge set, we introduce adjacency matrix to generation graph layer. The elements of the adjacency matrix indicate whether pairs of vertices are adjacent or not in the graph. If the token i is highly correlated with the intent j, the (i, j)-entry of the interactive graph corresponding adjacency matrix A will be assigned a value of 1.

To represent the vertex set, we concatenate the contextual representation and the intent embeddings as the output of this layer, denoting it as R :

$$R = [e_1, e_2, ..., e_n, i_1, i_2, ..., i_m] \tag{8}$$

Graph Attention Network over Intent-slot Interactive Graph

We use Graph Attention Networks(GAT) [Veličković *et al.*, 2018] to model intent-slot interactive graph. The input of j^{th} layer is an initial node features $R = \{r_1, r_2, ..., r_N\}$ $r_n \in \mathbf{R}^F$ and an adjacency matrix $A \in \mathbf{R}^{N\times N}$, where N denotes the number of the nodes and F is the the dimension of features at j^{th} layer. The output of j^{th} layer is a updated node features $R' = \{r'_1, r'_2, ..., r'_N\}$. The graph attention updating the representation of each node can be written as:

$$\mathcal{F}_k(r_i, r_j) = LeakyReLU(a^\top [W_h r_i \| W_h r_j]) \tag{9}$$

$$\alpha^k_{ij} = \frac{exp(\mathcal{F}_k(r_i, r_j))}{\sum_{j'\in\mathcal{N}_i} exp(\mathcal{F}_k(r_i, r_{j'}))} \tag{10}$$

$$r'_i = \|^K_{k=1}\sigma(\sum_{j\in\mathcal{N}_i} \alpha^k_{ij} W^k_h r_j) \tag{11}$$

where $W_h \in \mathbf{R}^{F'\times F}$, $a \in \mathbf{R}^{2F'}$ is the trainable weight matrix, $\mathcal{N}_i$ is the neighborhood of node i (including i) in the graph, α^k_{ij} is the attention coefficient computed by the k^{th} function $\mathcal{F}_k$, σ is nonlinear activation function, $\|$ denotes concatenation operation and K is the number of heads.

In this work, we adopt graph attention networks to model the slot-intent interactive graph. The input node features of GAT are matrix R, which is shown in Equation 8. The output node features are denoted as G.

$$G = GAT(R, A) \tag{12}$$

3.4 Slot Filling Decoder

For the slot-filling decoder, we similarly use a unidirectional LSTM as the slot filling decoder. At each decoding step t, the decoder state d_t is calculated by previous decoder state d_{t-1}, the previous emitted slot label distribution o^S_{t-1} and the aligned hidden state g_t:

$$d_i = LSTM(d_{i-1}, o^s_{i-1}, g_i) \tag{13}$$

Similarily, the decoder state d_i is utilized for slot filling

$$o^S_i = softmax(W_s d_i) \tag{14}$$

$$o^S_i = argmax(o^s_i) \tag{15}$$

where $o^S_i$is the predicted slot label of the i^{th} word in the utterance.

3.5 Joint Training

Following [Qin *et al.*, 2020], we adopt a joint model to update parameters by joint optimizing.

Intent Detection

The intent detection objective is:

$$\mathcal{L}_1 \triangleq -\sum_{k=1}^{N_I}(\widehat{o}^I_k log(o^I_k) + (1-\widehat{o}^I_k) log(1-o^I_k)) \tag{16}$$

where N_I is the number of single intent labels and $\widehat{o}^I_k$ is the gold intent label.

Slot Filling

Similarly, the slot filling task objective is defined as:

$$\mathcal{L}_2 \triangleq -\sum_{i=1}^{M}\sum_{j=1}^{N_S} \widehat{o}^{(j,S)}_i log(o^{(j,S)}_i) \tag{17}$$

where N_S is the number of slot labels and $\widehat{o}^S_k$ is the gold slot label.

Joint Training

To perform joint training for intent detection and slot filling, the final joint objective is formulated as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_2 \tag{18}$$

where λ is a hyper-parameter. During training, we set λ to 0.6.

4 Experiments

We conducted the experiments on several multiple intent and single intent datasets, comparing our method with a range of joint intent detect and slot filling baselines.

4.1 Datasets

Multiple Intent Datasets

We conducted the experiment on the MixSNIPS and MixATIS provided by [Qin *et al.*, 2020]. In these two datasets, the ratio of utterance containing 1-3 intents is [0.3,0.5,0.2]. There are 45,000 utterances for training, 2,500 utterances for validation and 2,500 utterances for testing on the MixSNIPS dataset. Similarly, MixATIS has 18,000 utterances for training, 1,000 for validation and 1,000 for testing.

Model	MixATIS				MixSNIP			
	Slot(F1)	Intent(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(F1)	Intent(Acc)	Overall(Acc)
Attention BiRNN	86.6	-	71.6	38.7	89.4	-	94.1	62.2
Slot-Gated	88.1	-	65.7	38.9	87.8	-	96.0	56.5
Slot-gated Intent	86.7	-	66.2	39.6	87.9	-	94.2	57.6
Bi-Model	85.5	-	72.3	39.1	86.8	-	95.3	53.9
SF-ID	87.7	-	63.7	36.2	89.6	-	96.3	59.3
Stack-Propagation	87.4	79.0	71.9	41.0	93.2	97.6	94.6	71.9
Joint Multiple ID-SF	87.5	80.6	73.1	38.1	91.0	98.2	95.7	66.6
AGIF	88.1	**81.2**	75.8	44.5	94.5	**98.6**	96.5	76.4
DGM	**88.7***	81.0	**76.7***	**47.1***	**94.7***	**98.6**	**96.7***	**78.0***

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

Single Intent Datasets

In addition, we also conducted experiments on two benchmark datasets, one is the widely-used ATIS dataset [Hemphill *et al.*, 1990] containing audio recordings of flight reservations, and the other is the custom-intent-engine dataset called the Snips [Coucke *et al.*, 2018], which is collected by Snips personal voice assistant.

4.2 Experimental Settings

Hyperparameters

In our experiments, we choose AGIF [Qin *et al.*, 2020] for developing the models. To prevent overfitting, we set dropout rate to 0.4. The graph layer number is 2. All embeddings are randomly initialized and fine-tuned during training. Our experiments are conducted on Tesla K80.

Baseline

We compared our model with the baselines including:

Attention BiRNN. Liu and Lane [2016] proposed the alignment-based RNN models with attention, which provide additional information to joint intent detection and slot filling.

Slot-Gated Atten. Goo et al. [2018] proposed a slot gate to learn the relationship between intent detection and slot filling.

Bi-Model. Wang et al. [2018] proposed Bi-model based RNN semantic frame to consider intent detection and slot filling cross impact.

SF-ID. Haihong et al. [2019] proposed an SF-ID network to establish connections for these two tasks. The iteration mechanism enhances the bi-directional interrelated connections.

Stack-Propagation Framework. Qin et al. [2019] adopted a joint model with Stack-Propagation which can use the token-level intent information as input for slot filling.

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

AGIF. Qin et al. [2020] proposed an Adaptive Graph-Interactive Framework for joint multiple intent detection and slot filling, which extracts the intents information for token-level slot prediction. This model achieves the state-of-the-art performance.

Model	Slot(F1)	Intent(Acc)	Overall(Acc)
Sentence-Level Aug	93.8	95.7	73.9
+Parameters	94.1	95.5	74.8
GCN-based	94.8	95.8	77.2
DGM	94.7	**96.7**	**78.0**

Table 2: Ablation Study on MixSNIPS Datasets.

4.3 Main Results

Following [Goo *et al.*, 2018] and [Qin *et al.*, 2019], we evaluated the performance of slot filling using F1 score, intent prediction using accuracy, the sentence-level semantic frame parsing using overall accuracy which represents slots and intent are both correctly-predicted in an utterance. Table 1 shows the experiment results of the proposed models on the MixSNIPS and MixATIS datasets.

Performance on Multiple Intent Datasets

We conducted experiments on two public Multiple-intent benchmarks. Table 1 shows the experiment results of the proposed models on the MixATIS and MixSNIPS datasets. On the MixATIS dataset, our model achieves 2.6% improvement in terms of Overall (Acc). On the MixSNIPS dataset, our model achieves 1.6% improvement on Overall. Results show that the intent-slot interactive graph constructed by our dynamic graph model can match the token to the relevant intent, so that our model can correctly predict the intent and slot in an utterance, thereby improving the overall accuracy.

We recorded the average runtime of our method(6,413s) and that of AGIF(33,528s) on MixATIS datasets. Measured from the runtime, DGM can increase the speed by about 5.2 times. The reason is that AGIF constructs a graph for each token in an utterance, while DGM constructs a graph for an utterance.

Performance on Single Intent Datasets

To prove the generalizability of our model, we conducted experiments on two public single-intent benchmarks. We compared our model with the single-intent state-of-the-art models including Joint Seq, Attention BiRNN, Slot-gated Atten, CAPSULE-NLU, SF-ID, Stack-Propagation. Table 3 shows the experiment results of the proposed models on SNIPS and

Model	ATIS			SNIP		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Joint Seq [Hakkani-Tür *et al.*, 2016]	94.3	92.6	80.7	87.3	96.9	73.2
Attention BiRNN [Bing and Lane, 2016]	94.2	91.1	78.9	87.8	96.7	74.1
Slot-Gated [Goo *et al.*, 2018]	94.8	93.6	82.2	88.8	97.0	75.5
Slot-gated Intent [Goo *et al.*, 2018]	95.2	94.1	82.6	88.3	96.8	74.6
Bi-Model [Wang *et al.*, 2018]	95.5	96.4	85.7	93.5	97.2	83.8
Self-Attentive Model [Li *et al.*, 2018]	95.1	96.8	82.2	90.0	97.5	81.0
CAPSULE-NLU [Zhang *et al.*, 2019]	95.2	95.0	83.4	91.8	97.3	80.9
SF-ID [Haihong *et al.*, 2019]	95.6	96.6	86.0	90.5	97.0	78.4
Stack-Propagation [Qin *et al.*, 2019]	95.9	96.9	86.5	94.2	98.0	86.9
DGM	**96.1***	**97.4***	**87.8***	**95.2***	**98.2***	**88.4***

Table 3: Slot filling and intent detection results on two single intent datasets. The numbers with * indicate that the improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test.

ATIS datasets. On the ATIS dataset, compared with the best prior joint model, our model achieves 1.3% improvement on Overall (Acc). On the SNIPS dataset, our model achieves 1.5% improvement on Overall (Acc). This indicates the effectiveness of our dynamic graph module.

4.4 Analysis

To further analyze the performance of the DGM model, we explored the effect of dynamic graph module. To better understand what the dynamic graph module has learned, we visualized intent attention weights of the intent-slot interactive graph, which is shown in Figure 3.

Effectiveness of Intent-slot Graph Interaction Mechanism

To verify the effectiveness of the dynamic graph module, we conducted experiments with the following ablations. We first conducted experiments by directly providing the same intent information for all tokens slot prediction where we concatenate the intent embedding and the hidden state of slot filling decoder. We refer to it as sentence-level augmented. We apply multiple LSTM layers (2-layers) to slot filling decoder and we name it as more parameters. The result is shown in the sentence-level augmented row of Table 2. From the result, we observe that the performance drops in all metrics in the MixSNIPS dataset. This indicates that the naive model of incorporating intent information decreases in the performance of the model.

Effectiveness of Graph Attention Mechanism

In another group of experiments, we adopted graph convolution layer instead of graph attention layer to model intent-slot interactive graph and keep other components unchanged. We refer to it as GCN-based model. The result is shown in the GCN-based row of Table 2. From the result, we observe that the performance drops in all metrics in the MixSNIPS dataset. This indicates that GAT can better encode intent-slot interactive graph, and the results of GCN-based also show the interactive graph can incorporate relevant intent information for each utterance.

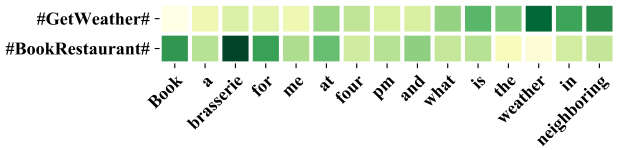

Figure 3: Visualization. Y-axis is the predicted intents and X-axis is the input utterance. For each column, the darker the color, the more relevant they are.

Visualization

To better understand what the dynamic graph model has learned, we visualized intent attention weights of the intent-slot interactive graph, which is shown in Figure 3. Based on the utterance "*Book a brasserie for me at four pm and what is the weather in neighboring*" and the intents "*BookRestaurant*" and "*GetWeather*", we can observe that our model properly attends the corresponding slot token "*brasserie*" and "*weather*" at intent "*BookRestaurant*"and "*GetWeather*" where the attention weights successfully focus on the correct slot, which means our model can capture the word that has played an important role in judging the intent of the utterance. This indicates that generation dynamic graph layer can leverage correct intent to construct the intent-slot interactive graph.

5 Conclusion

In this paper, we propose a dynamic sentence-level interactive graph model for joint multiple intent detection and slot filling, which enhances the directional interrelated connections for the two task. To further alleviate the error propagation, we design a novel method of constructing the graph, which boosts the speed by three to six times over the SOTA model. Experimental results on several multi-intent datasets show that our model achieves the SOTA performance.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key Research and Development Program of China [2016YFC0901902].

References

[Bing and Lane, 2016] L. Bing and I. Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016*, 2016.

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

[Gangadharaiah and Narayanaswamy, 2019] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North*, 2019.

[Goo *et al.*, 2018] Chih Wen Goo, Guang Gao, Yun Kai Hsu, Chih Li Huo, and Yun Nung Chen. 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)*, 2018.

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

[Haihong *et al.*, 2019] E. Haihong, P. Niu, Z. Chen, and M. Song. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

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

[Hemphill *et al.*, 1990] Charles T Hemphill, John J Godfrey, and George R Doddington. 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*, 1990.

[Hochreiter and Schmidhuber, 1997] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Li *et al.*, 2018] Changliang Li, Liang Li, and Ji Qi. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

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

[Qin *et al.*, 2020] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, November 2020.

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

[Sarikaya *et al.*, 2011] Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. Deep belief nets for natural language call-routing. In *2011 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5680–5683. IEEE, 2011.

[Tur and Mori, 2011] G. Tur and R. Mori. Spoken language understanding: Systems for extracting semantic information from speech. John Wiley and Sons, 2011.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2018] Yu Wang, Yilin Shen, and Hongxia Jin. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, June 2018.

[Xia *et al.*, 2018] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*, 2018.

[Xu and Sarikaya, 2013] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83. IEEE, 2013.

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

[Zhang *et al.*, 2019] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[Zhong *et al.*, 2018] Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, 2018.