A GRAPH ATTENTION INTERACTIVE REFINE FRAMEWORK WITH CONTEXTUAL REGULARIZATION FOR JOINTING INTENT DETECTION AND SLOT FILLING

Zhanbiao Zhu, Peijie Huang, Haojing Huang, Shudong Liu, Leyi Lao*

College of Mathematics and Informatics, South China Agricultural University, China

zbzhu@stu.scau.edu.cn, pjhuang@scau.edu.cn, {hjhuang, sudan, lylao}@stu.scau.edu.cn

ABSTRACT

Intent detection and slot filling are two important tasks for spoken language understanding. Considering the close relation between them, most existing methods joint them by sharing parameters or establishing explicit connection between them for potentially benefiting each other. However, most of them only consider single directional connection and ignore their cross-impact between them. Moreover, these joint methods treat the predicted labels as the gold labels, which may cause error propagation. In this paper, we propose a two-stage **G**raph **A**ttention **I**nteractive **R**efine (GAIR) framework. In stage one, the basic SLU model predicts the coarse intent and slots. In stage two, we select the top-k candidate labels from stage one and construct a graph to make full advantage of intent and slot filling information. By constructing such graph, our framework can establish a bidirectional connection between two tasks and refine the coarse result, which can better take full use of cross-impact between two tasks. Moreover, contextual regularization is introduced for better alleviating error propagation. Experiments on two datasets show that our model achieves the state-of-the-arts performance.

Index Terms— Spoken Language Understanding, Slot Filling, Intent Detection, Two-stage Refine Model

1. INTRODUCTION

Spoken language understanding (SLU) is a critical component in spoken dialogue systems, which aims to capture the semantics of user utterances or queries [1, 2]. It involves two tasks including intent detection and slot filling. Take a music-related utterance as an example, "我想听周杰伦的等你下课"(I want to listen to Jay Chou's Waiting For You). The intent of the utterance is "PlayMusic" and the slot labels for each word are {O, O, O, B-name, I-name, I-name, O, B-music_item, I-music_item,I-music_item, I-music_item}.

Considering the close relationship between intent detection and slot filling, some approaches [3–5] for joining intent detection and slot filling outperform pipeline models via mutual enhancement between two tasks by sharing model parameters. Recently, more and more works proposed joint intent detection and slot filling by using intent detection output to explicitly enhance slot filling. Goo et al. [6] and Li et al. [7] incorporated the intent information for the slot filling task by using the gate mechanism to control information flow from intent to slot. E et al. [8] designed an iteration mechanism to control predicted labels to mutually promote each other. Even though these models achieving promising results, most of them either joint two tasks by sharing parameters which is not interpretive or only consider the single-directional impact between intent detection and slot filling. What's more, joint methods [6–9] may cause error propagation, since they take the predicted labels as the gold labels.

For better taking advantage of the cross-impact between two tasks and alleviating error propagation, we propose a two-stage **G**raph **A**ttention **I**nteractive **R**efine (GAIR) framework. The framework takes top-k predicted labels in stage one as the candidate labels and constructs a graph with three different kinds of edges among token and intent, token and slot, intent and slot by establishing bidirectional connection, aiming to capture explicit intent and slot representation and make mutual interaction. However, directly introducing the candidate labels from stage one may damage the final performance of the model. To better improve the model's robustness, we construct different graphs via coarse predicted results and gold results to regularized utterance representation by graph attention network. Experimental results on two datasets, CAIS and SMP-ECDT show that our model outperforms compared models and achieves the state-of-the-arts performance. Codes for this paper are publicly available at https://github.com/BillKiller/GAIR-SLU.

2. APPROACH

This section describes the details of our model, as shown in Figure 1. We build our model based on state-of-the-art SLU model [9]. In stage one, the based model first generates the coarse intent detection and slot filling result. In stage two, we construct a graph based on the coarse top-k candidate results. The intent and slot filling correlated graph and the utterance representation will be fed in to refine module for better incorporating intent detection information and slot filling information by graph attention mechanism. For better alleviating er-

* Corresponding author.

ror propagation, we regularize the refine model by constraining candidate graph representation as close as possible to the gold context representation.

2.1. Basic Joint Model

In stage one, given a Chinese utterance $c = \{c_1, c_2, ..., c_N\}$with N characters, we feed the characters sequence into Stack-Propagation model to obtain the context representation, coarse intent prediction result and coarse slot filling result. The details can be seen in Stack-Propagation [10].

$$\boldsymbol{E}, \boldsymbol{Y^I}, \boldsymbol{Y^S} = StackPropagation(c_1, c_2, ..., c_N), \quad (1)$$

where $\boldsymbol{E} = \{\boldsymbol{e_1}, \boldsymbol{e_2}, ..., \boldsymbol{e_N}\}$ denote the context representation of the utterance encoded by the self-attentive [9, 11, 12] encoder, $\boldsymbol{Y^I} = \{\boldsymbol{y_1^I}, \boldsymbol{y_2^I}, ..., \boldsymbol{y_N^I}\}$ denote the output probability of intent labels for every token and $\boldsymbol{Y^S} = \{\boldsymbol{y_1^S}, \boldsymbol{y_2^S}, ..., \boldsymbol{y_N^S}\}$ means the output probability of the slot labels for every token. The candidate results are $\boldsymbol{o_i^I} = topK(\boldsymbol{y_i^I})$ and $\boldsymbol{o_i^S} = topK(\boldsymbol{y_i^S})$, where $\boldsymbol{o_i^I}$ and $\boldsymbol{o_i^S} \in \mathbb{R}^k$.

2.2. Graph Interaction Refine Module

In stage two, given the output probability of intent labels and slot filling tags, we construct a refine graph interaction layer so that the intent information can be propagated among word nodes, intent nodes and slot nodes. Graph neural networks were widely used in various NLP tasks for their capacity to model the structural information [13–16]. In particular, we utilize a graph attention network [17] which has been shown effective on SLU tasks to model the interaction between candidate intent labels and candidate slot labels at the token-level. The graph is constructed in the following way.

Vertices: To model the interaction between intent detection and slot filling task, we have three different nodes in the graph. Inspired by CM-Net [18], we have N_I intent memory $\boldsymbol{M^I} \in \mathbb{R}^{N_I \times d}$ and N_S slot memory $\boldsymbol{M^S} \in \mathbb{R}^{N_S \times d}$to initialize our node vertices where N_I represents the number of the intent labels and d represents the dimension of vertices representation. Besides, $\boldsymbol{E}$ is the utterance representation generated in stage one. Thus, we obtain the initialization node representation. $\boldsymbol{H^0} = [\boldsymbol{E}; \boldsymbol{M^S}; \boldsymbol{M^I}] \in \mathbb{R}^{(N+N_I+N_S)\times d}$.

Edges: There are three types of edges, we denote the graph adjacent matrix as $\boldsymbol{A} \in \mathbb{R}^{(N+N_I+N_S)\times(N+N_I+N_S)}$: **(a) intent connection:** we construct the intent connection where i-th token in the utterance connect to predicted top-k intent labels to explicitly leverage the candidate intent information: $A_{i,j} = 1$ (where $j \in o_i^I$, red lines in refine module in Figure 1). **(b) slot connection:** We construct the slot connection where i-th token in the utterance connects to predicted top-k slots labels to explicitly leverage the candidate slot information: $A_{i,j} = 1$(where $j \in o_i^S$, blue lines in refine module in Figure 1). **(c) intent-slot connection:** For the t-th token in the utterance, we establish the intent and slot edges by connecting

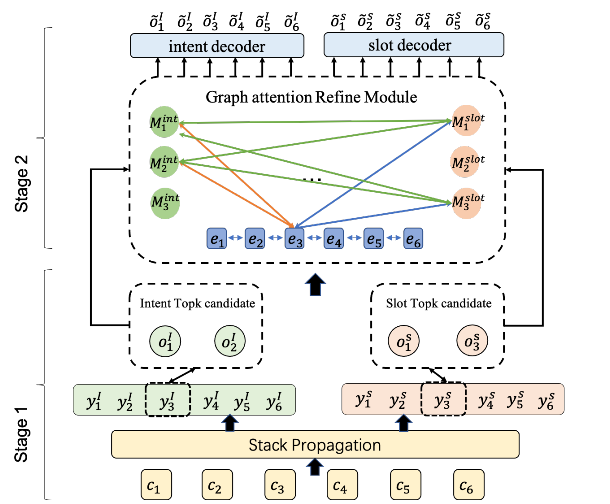

Fig. 1. The architecture of our model. Topk is the hyper-parameter of model. We use top 2 for demonstration.

candidate intent labels and slot tags: $A_{i,j} = 1$(where $i \in o_t^I$ and $j \in o_t^S$, green lines in refine module in Figure 1).

Vanilla Graph Attention Network. For a given graph with n nodes, GAT [17] takes node features $\tilde{\boldsymbol{H}} = \{\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, .., \tilde{\boldsymbol{h}}_n\}$ as input, and produce new representation $\tilde{\boldsymbol{H}}' = \{\tilde{\boldsymbol{h}}_1', \tilde{\boldsymbol{h}}_2', ..., \tilde{\boldsymbol{h}}_n'\}$. The attention mechanism of GAT can be written as:

$$\tilde{\boldsymbol{h}}_i' = ||_{k=1}^K \sigma(\textstyle\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \boldsymbol{W_h^k} \tilde{\boldsymbol{h}}_j), \quad (2)$$

$$\alpha_{ij} = \frac{exp(LeakyReLU(a^T[\boldsymbol{W_h}\tilde{\boldsymbol{h_i}}||\boldsymbol{W_h}\tilde{\boldsymbol{h_j}}]))}{\sum_{j'\in\mathcal{N}_i} exp(LeakyReLU(a^T[\boldsymbol{W_h}\tilde{\boldsymbol{h_i}}||\boldsymbol{W_h}\tilde{\boldsymbol{h_j}}]))}, \quad (3)$$

where K is the number of heads and α_{ij}^k is the attention weight at k head. We use $||$ to concatenate all the heads. $\mathcal{N}_i$ denotes the neighbors of node i (including i)

2.3. Refine Decoder

For intent detection and slot filling decoder, we use a unidirectional LSTM [19, 20] as the decoder. The input of refine decoder is $\tilde{\boldsymbol{H}}^{[1,N]'} = \{\tilde{\boldsymbol{h}}_1', ..., \tilde{\boldsymbol{h}}_N'\}$. At the decoding step i, the slot decoder hidden state $\boldsymbol{h_i}$ can be formalized as: $\boldsymbol{h_i^S} = f(\boldsymbol{h_{i-1}^S}, \boldsymbol{y_{i-1}^S}, \tilde{\boldsymbol{h}}_i')$ and the intent decoder hidden state can be formalized as: $\boldsymbol{h_i^I} = f(\boldsymbol{h_{i-1}^I}, \boldsymbol{y_{i-1}^I}, \tilde{\boldsymbol{h}}_i')$, where $\boldsymbol{h_{i-1}^S}$ and $\boldsymbol{h_{i-1}^I}$ are the previous decoder hidden state for slot filling decoder and intent detection decoder. $\tilde{\boldsymbol{y}}_{i-1}^S$ and $\tilde{\boldsymbol{y}}_{i-1}^I$ are the previous emitted slot label and intent label distribution. the hidden state is used for intent detection and slot filling. $\tilde{\boldsymbol{y}}_i^I = softmax(\boldsymbol{W_h^I}\boldsymbol{h_i^I})$ and $\tilde{\boldsymbol{y}}_i^S = softmax(\boldsymbol{W_h^S}\boldsymbol{h_i^S})$. $\tilde{o}_i^I = argmax(\tilde{\boldsymbol{y}}_i^I)$ and $\tilde{o}_i^S = argmax(\tilde{\boldsymbol{y}}_i^S)$ represents the predicted intent labels and slot tags, where $\boldsymbol{W_h^S}$ and $\boldsymbol{W_h^I}$ are trainable parameters. We use most frequent intent labels by voting from all tokens as the final intent label. For better discrimination, $\boldsymbol{y_i^I}$ and $\boldsymbol{y_i^S}$ represent the output probability in stage one, while $\tilde{\boldsymbol{y}}_i^{\boldsymbol{I}}$ and $\tilde{\boldsymbol{y}}_i^{\boldsymbol{S}}$ are in stage two.

2.4. Contextual Regularization

Since we select top-k candidate labels to construct the graph, noise may be introduced into the refine module. We introduce contextual regularization to constraint error propagation and improve model adaptation. During the training process, we use the gold label to construct the graph same as above, and GAT model generates the gold utterance representation: $\tilde{\boldsymbol{H}}_{\boldsymbol{pos}}^{'} = GAT(\boldsymbol{A_{pos}}, \tilde{\boldsymbol{H}})$, where $\boldsymbol{A_{pos}}$ is the adjacent matrix of the graph constructed by gold labels. The utterance representation by candidate graph should as be close as possible to the gold utterance representation by gold graph. Thus, we minimize the regularization loss functions by using Mean Square Error (MSE) for the gold utterance representation and top-k candidate utterance representation, which can be formulated as follows:

$$\mathcal{L}_t = MSE(\tilde{\boldsymbol{H}}_{\boldsymbol{pos}}^{'}, \tilde{\boldsymbol{H}}^{'}). \tag{4}$$

2.5. Multi-task learning

Following Goo et al. [6] and Qin et al. [9], we adopt a joint training model to consider the two-tasks. The coarse results of intent detection and slot filling from stage one provide candidate results for refine module. The joint objection of intent detection and slot filling in stage one is formulated as:

$$\mathcal{L}_{c\_I} \triangleq -\sum_{t=1}^{T}\sum_{j=1}^{n_I} \hat{y}_j^{t,I} log(y_j^{t,I}), \tag{5}$$

$$\mathcal{L}_{c\_S} \triangleq -\sum_{t=1}^{T}\sum_{j=1}^{n_S} \hat{y}_j^{t,S} log(y_j^{t,S}), \tag{6}$$

where $\hat{y}$ is the gold label.

The refine joint objection in stage two is formulated as:

$$\mathcal{L}_{I} \triangleq -\sum_{t=1}^{T}\sum_{j=1}^{n_I} \hat{y}_j^{t,I} log(\tilde{y}_j^{t,I}), \tag{7}$$

$$\mathcal{L}_{S} \triangleq -\sum_{t=1}^{T}\sum_{j=1}^{n_S} \hat{y}_j^{t,S} log(\tilde{y}_j^{t,S}). \tag{8}$$

The final joint objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{c\_I} + \mathcal{L}_{c\_S} + \mathcal{L}_I + \mathcal{L}_S + \mathcal{L}_t. \tag{9}$$

3. EXPERIMENTS

3.1. Experiment Setting

To evaluate the effectiveness of our model, we conduct experiments on two publicly available Chinese SLU datasets, CAIS [18] and SMP-ECDT[1] [10]. The CAIS dataset includes 7,995 training, 994 validation and 1024 test utterances. Because only training data is provided in SMP-ECDT, we split a new valid set and test set. The SMP-ECDT dataset includes 1656 training, 414 validation and 509 test utterances.

The compared models are *Slot-Gated Full Atten* [6], *SF-ID Network* [8], *CM-Net* [18], *Stack-Propagation* [9], *MLWA* [10]. On CAIS dataset, we use the reported performances of these models from literature [10]. On SMP-ECDT, using the split test set, we run the published code of the compared models, except for *CM-Net* [18] which does not provide codes.

We utilize *F1 Score* and *Accuracy* to evaluate slot filling and intent detection respectively. The sentence-level semantic frame parsing using *Overall accuracy* which denotes the both intent and slot filling are correct in an utterance. In this paper, the head number of the graph is 4, the hidden dimension of the graph attention is 128. The dropout [21] rate is 0.4. We use Adam [22] to optimize the parameters of the model. For all the experiments, we select the model which works the best on the dev set and then evaluate it on the test set, and we conduct our experiments with ten different random seeds.

3.2. Main Results

From the result in Table 1 we have the following observations: (1) Our model outperforms *Stack-Propagation* on all metrics especially in Slot(F1) and Overall(Acc). Our framework outperform *Stack-Propagation* by 1.28% on Slot(F1) and 1.08% on Intent(Acc) on CAIS. Besides, we achieves 4.54% and 0.15% improvements on Slot(F1) and Intent(Acc) on SMP-ECDT. Furthermore, our model can take cross-impact between two tasks into consideration and enhance each other simultaneously which can improve sentence-level semantic frame parsing. For this reason, our model gains 2.18% and 3.41% improvements on Overall(Acc) for CAIS and SMP-ECDT and achieve state-of-the-art performance. (2) Because *MLWA* model used extra languish knowledge to enhance model performance, their results are not directly comparable with ours. But our model still outperforms *MLWA* on most metrics. On CAIS dataset, the performances of our model are improved on all the metrics. On SMP-ECDT, we achieve a large margin improvement(+3.58%) on Slot(F1) and 0.2% improvement on Overall(Acc).

3.3. Analysis

Effect on Intent Connection and Slot Connection. We remove the intent related connection in the graph which means that we only use the slot candidate information explicitly. We name it as *without intent connection*. The results are shown in Table 2, the performance of intent detection and slot filling drops, especially in slot filling task which is highly guided by intent information [6, 9]. Similarly, we remove the slot related connection in the refine-layer. We conduct *without slot connection* experiment. The result shows that slot filling and intent detection drop. It demonstrates that intent and slot representations are critical to the both tasks.

[1] http://conference.cipsc.org.cn/smp2019/evaluation.html

Model	CAIS Dataset			SMP-ECDT Dataset		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Slot-Gated Full Atten [6]	81.13	94.37	80.83	60.88	84.81	54.21
SF-ID Network [8]	84.85	94.27	82.41	64.52	86.51	56.33
CM-Net [18]	86.16	94.56	-	-	-	-
Stack-propagation [9]	87.64	94.37	84.68	71.62	88.65	63.78
MLWA † [10]	88.61	95.16	86.17	72.58	90.37	66.99
GAIR(stage one)	88.68	95.35	84.98	74.76	**88.99**	66.90
GAIR	**88.92***	**95.45***	**86.86***	**76.16***	88.80	**67.19***

Table 1. SLU performance on CAIS and SMP-ECDT datasets. † their result are not directly comparable with ours, since they used extra segmentation information. The numbers with * indicate that the improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test.

Model	CAIS Dataset			SMP-ECDT Dataset		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
without Intent Connection	88.15	94.86	85.07	74.61	88.21	62.86
without Slot Connection	88.41	95.26	85.27	74.45	87.03	62.47
without Contextual Regularization	85.65	95.16	83.10	71.80	87.22	61.29
GAIR	88.92	95.45	86.86	76.16	88.80	67.19

Table 2. Ablation experiments on the CAIS and SMP-ECDT datasets.

Effect on Contextual Regularization. We conduct *without Contextual Regularization* experiment. We observe that 3.27% drops on Slot(F1) and 0.29% drops on Intent(Acc) on CAIS. Besides, we observe 4.36% and 1.58% drops on Slot(F1) and Intent(Acc) on SMP-ECDT. We attribute this to the fact that introducing gold labels to regularize model do help model alleviating error propagation and improve the adaptation ability and robustness.

Effect on Refine Module. From Table 1, we can observe that the performance of our model on all metrics in stage two outperforms stage one which shows that refine module do take coarse results into consideration and refine both tasks. The refine module can correct the error prediction. Taking an example "播放郁可唯的歌"(play the song of Kewei Yu) for case study, our model predicted "郁可唯"(Kewei Yu) as the name of a song in stage one, and after feeding into refine module, our model refined the incorrect slot tags and correctly predicted the slots of "郁可唯" as an artist. Besides, the performance of our model in stage one outperforms *Stack-propagation*, which demonstrate the refine module not only improves the final result but also improves the coarse result in stage one during back-propagation.

4. RELATED WORK

Intent Detection and Slot Filling. Considering the strong correlation between intent detection and slot filling. Dominant SLU systems adopt joint models, which can be classified into two categories. One is implicit joint modeling, which only adopts a shared encoder to capture the shared features [3]. Another category is explicit joint modeling. It explicitly controls the process of interaction, including the Slot-Gated model proposed by Goo *et al.* [6], the Stack-Propagation model proposed by Qin *et al.* [9], the CM-Net introduced by Liu *et al.* [18] and a Co-Interactive Transformers [11] proposed by Qin *et al.* [23]. Our model considers the joint method by Graph Attention Network(GNN).

Graph Neural Network for NLP. Graph neural networks were widely used in various NLP tasks for its capacity to model the structural information. Hu *et al.* [14] and Huang *et al.* [15] applied the graph attention network [17] for classification tasks to incorporate the dependency parser information. Graph structure is also successfully conducted on modeling the discourse information for the summarizing generation task. Feng *et al.* [16] successfully applied a graph network for dialogue summarization. In SLU, Qin *et al.* [13] applied the graph neural network to guide slot filling by using the intent information. In our work, we apply the graph network to build the relationship between the hidden of token, slots and intent to refine its performance.

5. CONCLUSION

In this paper, we proposed a graph attention interaction refine framework with contextual regularization mechanism for jointing candidate intent labels and slot filling tags which can consider cross-impact between two tasks and refine coarse results from stage one. Experimental results on two datasets show that our framework outperforms compared models and achieves the state-of-the-arts performance.

6. ACKNOWLEDGEMENTS

This work was supported by Natural Science Foundation of Guangdong Province (No.2021A1515011864), National Natural Science Foundation of China (No.71472068), Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation (No.pdjh2022a007) and Innovation Training Project for College Students of Guangdong Province (No.S202110564051).

7. REFERENCES

[1] Gokhan Tur and Renato De Mori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, Wiley, New York, 2011.

[2] Henry Weld, Xiaoqi Huang, Siqi Long, Josiah Poon, and Soyeon Caren Han, "A survey of joint intent detection and slot-filling models in natural language understanding," *CoRR*, vol. abs/2101.08091, 2021.

[3] Bing Liu and Ian Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proc. of INTERSPEECH 2016*, pp. 685–689.

[4] Yu Wang, Yilin Shen, and Hongxia Jin, "A bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *Proc. of NAACL-HLT 2018*, pp. 309–314.

[5] Xiaodong Zhang and Houfeng Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *Proc. of IJCAI 2016*, pp. 2993–2999.

[6] Chih-Wen Goo, Guang Gao, and Yun-Kai Hsu, et al., "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. of NAACL-HLT 2018*, 2018, pp. 753–757.

[7] Changliang Li, Liang Li, and Ji Qi, "A self-attentive model with gate mechanism for spoken language understanding," in *Proc. of EMNLP 2018*, pp. 3824–3833.

[8] Haihong E, Peiqing Niu, and Zhongfu Chen, et al., "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proc. of ACL 2019*, pp. 5467–5471.

[9] Libo Qin, Wanxiang Che, and Yangming Li, et al., "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proc. of EMNLP-IJCNLP 2019*, pp. 2078–2087.

[10] Dechuan Teng, Libo Qin, Wanxiang Che, Sendong Zhao, and Ting Liu, "Injecting word information with multi-level word adapter for Chinese spoken language understanding," in *Proc. of ICASSP 2021*, pp. 8188–8192.

[11] Ashish Vaswani, Noam Shazeer, and Niki Parmar, et al., "Attention is all you need," in *Proc. of NIPS 2017*, pp. 5998–6008.

[12] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi, "Deep semantic role labeling with self-attention," in *Proc. of AAAI 2018*, pp. 4929–4936.

[13] Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu, "GL-GIN: fast and accurate non-autoregressive model for joint multiple intent detection and slot filling," in *Proc. of ACL-IJCNLP 2021*, pp. 178–188.

[14] Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proc. of EMNLP-IJCNLP 2019*, pp. 4820–4829.

[15] Binxuan Huang and Kathleen M. Carley, "Syntax-aware aspect level sentiment classification with graph attention networks," in *Proc. of EMNLP-IJCNLP 2019*, pp. 5468–5476.

[16] Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng, "Dialogue discourse-aware graph model and data augmentation for meeting summarization," in *Proc. of IJCAI 2021*, pp. 3808–3814.

[17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph attention networks," in *Proc. of ICLR 2018*.

[18] Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu, "CM-Net: A novel collaborative memory network for spoken language understanding," in *Proc. of EMNLP-IJCNLP 2019*, pp. 1051–1060.

[19] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Kaisheng Yao, Baolin Peng, and Yu Zhang, et al., "Spoken language understanding using long short-term memory neural networks," in *Proc. of SLT 2014*, pp. 189–194.

[21] Nitish Srivastava, Geoffrey E. Hinton, and Alex Krizhevsky, et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR 2015*.

[23] Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu, "A co-interactive transformer for joint slot filling and intent detection," in *Proc. of ICASSP 2021*, pp. 8193–8197.