LEVERAGING BILINEAR ATTENTION TO IMPROVE SPOKEN LANGUAGE UNDERSTANDING

*Dongsheng Chen, Zhiqi Huang, Yuexian Zou**

ADSPLAB, School of ECE, Peking University, Shenzhen, China

ABSTRACT

Spoken language understanding system (SLU) typically includes two tasks: Intent detection (ID) and Slot filling (SF). Optimizing these two tasks in an interactive way with attention mechanism has been shown effective. However, previous attention-based works leveraged only the first order attention design, which is lacking in efficacy. To trigger more adequate information interaction between the input intent or slot features, we propose a novel framework with Bilinear attention, which can build the second order feature interactions. By stacking numerous Bilinear attention modules and equipping the Exponential Linear Unit activation, it can build higher and infinity order feature interactions. To demonstrate the effectiveness of the proposed framework, we conduct some experiments on two benchmark datasets, i.e., SNIPS and ATIS. And the experimental results show that our framework is more competitive than multiple baselines as well as the first order attention model.

Index Terms— Spoken Language Understanding, Bilinear Attention, Features Interaction, Multitask Learning

1. INTRODUCTION

Intent detection (ID) and Slot filling (SF) play important roles in SLU system. An example of the SLU task is shown in Figure 1, given an utterance "*I want to listen to Sleep Alone*", ID can be seen as a classification task to identity the user's intent is to listen to a song and SF can be treated as a sequence labeling task to predict slot labels in BIO format [1] which demonstrates that "*Sleep Alone*" is the song's title.

Though ID and SF tasks can be considered separately, they have a strong correlation, e.g., once we know that the user's intent is *"Play Song"*, it would be easier for us to predict the slot labels of *"Sleep Alone"*, and vice versa. Thus, many previous works proposed to jointly model the ID and SF tasks for better performance [2–4], showing that Attention-based methods [5–8] are effective to leverage the attention mechanism [9] to trigger the mutual interaction between intent features and slot features. Specifically, the attention mechanism learns a set of weights to reflect the importance of different words of an utterance via linearly fusing the given query and key by element-wise sum, the weights are then applied to the values to obtain a weighted sum which represents the enhanced intent or slot features in a mutual way.

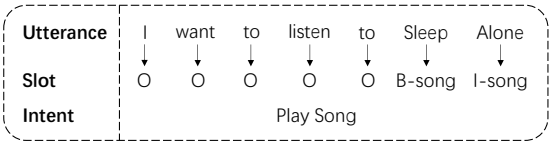

Fig. 1. Utterance with intent and slot labels in BIO format.

The *Co-Interactive* model proposed by [8] triggers the mutual interaction between intent and slot features via attention, and achieved the state-of-the-art results. In this paper, we argue that the inherent design of conventional attention mechanism can only model the first order feature interactions and is inefficient for SLU task. To explore higher order feature interactions, we leverage bilinear pooling [10] to build the second order interactions, which is an operation to calculate outer product between two vectors. By taking all pairwise interactions between query and key into account, it triggers the second order feature interactions, and thus provides more discriminative representations, which has been widely explored in Computer Vision research [11–14].

As shown in Figure 2, we introduce a Bilinear attention module to build the second order interactions between intent and slot features, and it is used to replace the conventional attention. Stacking an appropriate number of such modules is easy to group to outperform the bilinear models and extract higher order feature interactions. We also provide theoretical analysis on that the model can build infinity order feature interactions by equipping with Exponential Linear Unit (ELU) [15]. Finally, the experimental results on two datasets SNIPS [16] and ATIS [17] show that our framework illustrated in Figure 3 outperforms previous methods, which proves the effectiveness of our approach.

To summarize, the contributions of this work are as follows: 1) We leverage Bilinear attention to build higher order attention-based framework, which can better trigger features interaction between ID and SF to improve SLU task; 2) We give an elegant framework of how the BiLinear attention block could be extended for building higher or even infinity order interactions; 3) The experimental results show that the

*Corresponding author: zouyx@pku.edu.cn

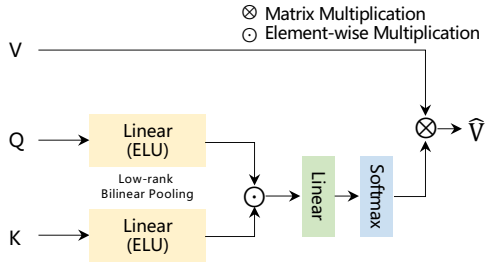

Fig. 2. Overview of Bilinear attention.

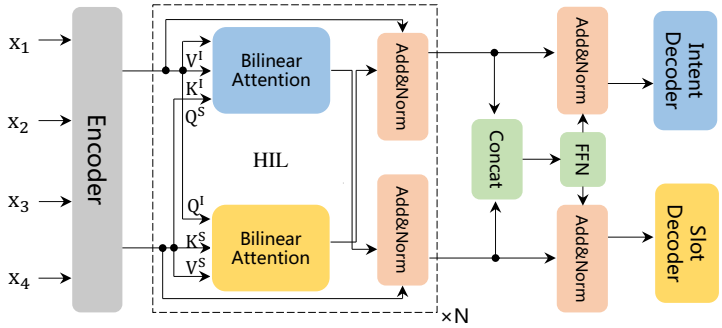

Fig. 3. Architecture of the proposed framework with Bilinear attention module.

proposed framework achieves better results consistently compared to previous SLU models as well as the first order attention model on two benchmark datasets.

2. METHODS

In this section, we first formulate the proposed Bilinear attention module, and then introduce our proposed framework briefly.

2.1. Bilinear attention

The Bilinear attention module is shown in Figure 2. Consider that there is a query $\mathbf{q} \in \mathbb{R}^d$, a set of keys $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^n$, and a set of values $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^n$, where $\mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^d$ denote the i-th key/value pair. A joint bilinear query-key representation $\mathbf{B}_{ki} \in \mathbb{R}^d$ is computed to build the second order feature interactions between query and key through the low-rank bilinear pooling operation:

$$\mathbf{B}_{ki} = (\mathbf{W}_{qk}\mathbf{q}) \odot (\mathbf{W}_k\mathbf{k}_i), \tag{1}$$

where $\mathbf{W}_{qk}, \mathbf{W}_k \in \mathbb{R}^{d\times d}$ are two weight matrices, $\odot$ denotes element-wise multiplication.

Next, the contextual bilinear attention distribution is introduced to aggregate the contextual information within all values by using all bilinear query-key representations $\{\mathbf{B}_{ki}\}_{i=1}^n$. Specifically, the contextual bilinear attention distribution is introduced by using a softmax layer to normalize each bilinear query-key representation into the corresponding attention weight:

$$b_i = \mathbf{W}_b\mathbf{B}_{ki}, \beta = \text{softmax}(\mathbf{b}), \tag{2}$$

where $\mathbf{W}_b \in \mathbb{R}^{1\times d}$, b_i is the i-th element in $\mathbf{b}$, and β_i denotes the normalized contextual attention weight for the i-th key/value pair.

Then finally, the Bilinear attention module outputs the enhanced value feature $\hat{\mathbf{v}}_i$ by accumulating the i-th values $\mathbf{v}_i$ with contextual bilinear attention weights:

$$\hat{\mathbf{v}}_i = \sum_{i=1}^{n} \beta_i \mathbf{v}_i. \tag{3}$$

As such, our Bilinear attention module produces more representative attended features since the second order feature interactions are exploited via bilinear pooling. We repeat the above process n times with $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^n$, and get a set of values $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_i\}_{i=1}^n$, we denote it as:

$$\hat{\mathbf{V}} = \text{Att}_{bilinear}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \tag{4}$$

where $\hat{\mathbf{V}} \in \mathbb{R}^{n\times d}$ is the enhanced feature with the second order attention feature interactions. To get higher order feature interactions, we further iterate the above process by using a stack of Bilinear attention modules, i.e., we can get $2N^{th}$ order feature interactions by repeating the process N times:

$$\hat{\mathbf{V}^{\mathbf{N}}} = \text{Att}_{bilinear}(\mathbf{Q}^{\mathbf{N}}, \mathbf{K}^{\mathbf{N}}, \mathbf{V}^{\mathbf{N}}). \tag{5}$$

It can model infinity order interactions by equipping the module with Exponential Linear Unit (ELU) [15] on Eq. (1), which can be proved via Taylor expansion of each element in bilinear vector after exponential transformation. Specifically, for two vectors A and B, their exponential bilinear pooling can be estimated using the Taylor expansion:

$$\begin{aligned}
&\exp\left(\mathrm{W}_A\mathrm{A}\right) \odot \exp\left(\mathrm{W}_B\mathrm{B}\right) \\
&= \left[\exp\left(\mathrm{W}_A^1\mathrm{A}\right) \odot \exp\left(\mathrm{W}_B^1\mathrm{B}\right), \ldots, \exp\left(\mathrm{W}_A^D\mathrm{A}\right) \odot \exp\left(\mathrm{W}_B^D\mathrm{B}\right)\right] \\
&= \left[\exp\left(\mathrm{W}_A^1\mathrm{A} + \mathrm{W}_B^1\mathrm{B}\right), \ldots, \exp\left(\mathrm{W}_A^D\mathrm{A} + \mathrm{W}_B^D\mathrm{B}\right)\right] \\
&= \left[\sum_{p=0}^{\infty} r_p^1 \left(\mathrm{W}_A^1\mathrm{A} + \mathrm{W}_B^1\mathrm{B}\right)^p, \ldots, \sum_{p=0}^{\infty} r_p^D \left(\mathrm{W}_A^D\mathrm{A} + \mathrm{W}_B^D\mathrm{B}\right)^p\right],
\end{aligned}$$

where W_A and W_B are weight matrices, D denotes the dimension of bilinear vector, $\mathrm{W}_A^i/\mathrm{W}_B^i$ is the i-th row in $\mathrm{W}_A/\mathrm{W}_B$.

2.2. Higher order Interaction framework

As can be seen in Figure 3, the Higher order Interaction framework consists of four parts, i.e., Utterance encoder, Higher order Interaction Layer (HIL), Feature Fusion Layer (FFL), and Decoder. We will describe them in the following sections.

2.2.1. Utterance Encoder

We use a shared BiLSTM [18] as the Utterance Encoder to leverage the advantages of temporal features within word orders and contextual information. Given the input utterance $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, where n is the number of tokens, the BiLSTM produces a series of context-aware hidden states $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$. And the $\mathbf{H}$ is used as both intent and slot features for convenience here, i.e., $\mathbf{H_I} = \mathbf{H_S} = \mathbf{H}$.

2.2.2. Higher order Interaction Layer (HIL)

The $\mathbf{H_I}$ and $\mathbf{H_S}$ are further fed into the HIL to strengthen both the intent and slot features by capturing higher order feature interactions between them.

Formally, the HIL is composed of a stack of N identical sublayers, which consists of Bilinear attention module and layer normalization [19]. We first map the matrix $\mathbf{H_I}$ and $\mathbf{H_S}$ to queries($\mathbf{Q_I}, \mathbf{Q_S}$), keys($\mathbf{K_I}, \mathbf{K_S}$) and values($\mathbf{V_I}, \mathbf{V_S}$) matrices by different learnable linear projections. Then we take $\mathbf{Q_I}$, $\mathbf{K_S}$ and $\mathbf{V_S}$ as queries, keys and values, respectively, acquiring the enhanced values:

$$\begin{aligned}\hat{\mathbf{V}}_\mathbf{I} =& \mathrm{Att}_{bilinear}(\mathbf{Q_I}, \mathbf{K_S}, \mathbf{V_S}),\\ \mathbf{H_I} =& \mathrm{LayerNorm}(\hat{\mathbf{V}}_\mathbf{I} + \mathbf{H_I}),\end{aligned} \tag{6}$$

similarly, we take $\mathbf{Q_S}$ as queries, $\mathbf{K_I}$ as keys and $\mathbf{V_I}$ as values to obtain $\mathbf{H_S}$. After iterating N times, we obtain the enhanced intent features $\mathbf{H_I^N} \in \mathbb{R}^{n\times d}$ and slot features $\mathbf{H_S^N} \in \mathbb{R}^{n\times d}$, where the higher $2N^{th}$ order interactions are triggered between the two types of features.

2.2.3. Feature Fusion Layer (FFL)

Inspired by [20,21], we leverage Position-wise Feed-Forward Networks (FFN) to fuse intent and slot features implicitly. First, we concatenate $\mathbf{H_I^N}$ and $\mathbf{H_S^N}$ to combine the intent and slot information:

$$\mathbf{H_{IS}} = [\mathbf{H_I^N}, \mathbf{H_S^N}], \tag{7}$$

where $[\cdot, \cdot]$ indicates concatenation. Then, we leverage a shared FFN followed by layer normalization to acquire the updated intent features $\hat{\mathbf{H}_\mathbf{I}}^\mathbf{N} \in \mathbb{R}^{n\times d}$ and slot features $\hat{\mathbf{H}_\mathbf{S}}^\mathbf{N} \in \mathbb{R}^{n\times d}$, i.e.:

$$\begin{aligned}\hat{\mathbf{H}_\mathbf{I}}^\mathbf{N} &= \mathrm{LayerNorm}(\mathrm{FFN}(\mathbf{H_{IS}}) + \mathbf{H_I^N}),\\ \hat{\mathbf{H}_\mathbf{S}}^\mathbf{N} &= \mathrm{LayerNorm}(\mathrm{FFN}(\mathbf{H_{IS}}) + \mathbf{H_S^N}).\end{aligned} \tag{8}$$

2.2.4. Decoder

Intent Decoder We apply maxpooling operation on $\hat{\mathbf{H}_\mathbf{I}}^\mathbf{N}$ to obtain utterance representation $\boldsymbol{e}$, which is used as input to predict the intent label:

$$\begin{aligned}\hat{\mathbf{y}}^\mathbf{I} &= \mathrm{softmax}\left(\mathbf{W}^\mathbf{I}\boldsymbol{e} + \mathbf{b_I}\right),\\ \mathbf{o}^\mathbf{I} &= \mathrm{argmax}(\hat{\mathbf{y}}^\mathbf{I}),\end{aligned} \tag{9}$$

where $\hat{\mathbf{y}}^\mathbf{I}$ is the output intent distributation, $\mathbf{o}^\mathbf{I}$ represents the predicted intent label, $\mathbf{W}^\mathbf{I}$ and $\mathbf{b_I}$ are learnable parameters.

Slot Decoder We use a standard CRF layer [3] to build the dependency between labels:

$$\begin{aligned}\mathbf{O_S} &= \mathbf{W}^\mathbf{S}\hat{\mathbf{H}_\mathbf{S}}^\mathbf{N} + \mathbf{b_S},\\ P(\hat{y}|\mathbf{O_S}) &= \frac{\sum_{i=1} \exp f(y_{i-1}, y_i, \mathbf{O_S})}{\sum_{y'}\sum_{i=1} \exp f(y'_{i-1}, y'_i, \mathbf{O_S})},\end{aligned} \tag{10}$$

where $f(y'_{i-1}, y'_i, \mathbf{O_S})$ computes the transition score from y'_{i-1} to y'_i, and $\hat{y}$ represents the predicted label sequence.

3. EXPERIMENTS

3.1. Datasets

To evaluate the validity of our proposed approach, we conduct experiments on two benchmark datasets[1], SNIPS [16] and ATIS [17]. Both evaluated datasets used in our paper follow the same format and partition as in [2] and are the same as most previous works.

3.2. Experimental Settings

In the framework, the hidden dimensionality d is set as 128, and the sublayer number N of HIL is set as 2. We use 300d GloVe pre-trained vector [24] as the initialization embedding. We adopt Adam [25] optimizer for the parameters updating, with a batch size of 32 and initial learning rate of 0.001. Following [7], three evaluation metrics are used for the SLU task, i.e., the intent accuracy, the slot F1 score, and the overall accuracy. The model which works the best on the dev set will be chosen, and then we evaluate it on the test set.

3.3. Experimental Results and Analysis

3.3.1. Main Results

From Table 1, we can see that our framework significantly outperforms all the baselines, which outperforms the state-of-the-art model on SNIPS and ATIS by 0.12% and 0.53% in terms of accuracy on ID task, 0.57% and 0.22% in terms of F1 score on SF task, 0.93% and 1.06% in terms of overall accuracy. It demonstrates the effectiveness of capturing higher order feature interactions between intents and slots via our Bilinear attention in our framework.

[1] https://github.com/MiuLab/SlotGated-SLU/tree/master/data/

Table 1. SLU Performance evaluation results. Our framework outperforms baselines on both datasets under t-test ($p<0.05$).

Model	SNIPS			ATIS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
SF-ID Network [22]	90.46	97.00	78.37	95.58	96.58	86.00
Stack-Propagation [7]	94.20	98.00	86.90	95.90	96.90	86.50
Graph-LSTM [23]	95.30	98.29	89.71	95.91	97.20	87.57
Co-Interactive Transformer [8]	95.90	98.80	90.30	95.90	97.70	87.40
Baseline (BiLSTM + Decoder)	94.19	97.79	85.86	95.32	95.63	84.99
+ FFL	94.84	97.86	87.19	95.49	96.42	85.78
+ Conventional attention	95.18	98.27	88.05	95.64	97.39	86.81
+ Bilinear attention	95.75	98.43	88.84	95.77	97.12	87.28
+ ELU	95.92	98.67	89.57	96.00	97.31	87.67
Our framework	**96.47**	**98.92**	**91.23**	**96.12**	**98.23**	**88.46**

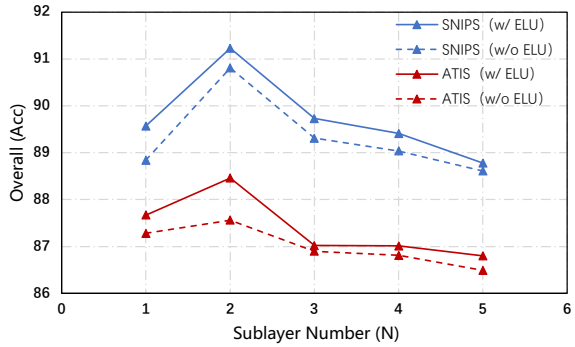

Fig. 4. Effect of HIL's sublayer number N on the model's performance.

3.3.2. Ablation Studies

Ablation studies results are shown in the second set of Table 1. From the baseline model (BiLSTM + Decoder), we can see that as adding each key component of our framework gradually, the performance becomes better, and reaches the best when equipping with ELU (3.71% and 2.68% absolute improvement compared to baseline in overall accuracy on the SNIPS and ATIS dataset, respectively). The performance improves gradually demonstrates the positive effect of each key component, and the improvement brought by ELU proves the core advantage of exploiting higher and even infinity order feature interactions between intent and slot.

3.3.3. Effect of sublayer number in HIL

Considering that too many Bilinear attention modules can make the model overfits and somewhat hinder the exploitation of higher order features interaction, we designed experiments on both datasets by varying the HIL's sublayer number from one to five and remaining other components unchanged, in order to find the best number of sublayers in HIL for the whole framework. The results is illustrated in Figure 4.

We can see that the performance of the model equipped with ELU will be better than the one without ELU in any case. When the number of sublayers in HIL is two, the model's performance on both datasets gets the best. So we set the sublayer number in HIL as two ($N = 2$) mentioned in section 3.2. When the sublayer number is more than two, the performance gets worse.

4. CONCLUSION

In this paper, we propose a novel framework for spoken language understanding with Bilinear attention module involved to build second and higher order features interaction between intent detection and slot filling tasks. Through this module, more discriminative intent and slot representations can be built. Besides, we demonstrate that when stacking an appropriate amount of Bilinear attention modules, or equipping modules with ELU activation, higher and even infinite order feature interactions are established between intent and slot. Experiments on two benchmark datasets show the effectiveness of the proposed models. In the future, we will continue to explore more properties of Bilinear attention (e.g., Robustness) to improve other existing SLU models.

5. ACKNOWLEDGEMENTS

This research was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: GXWD20201231165807007-20200814115301001). Special acknowledgements are given to AOTO-PKUSZ Joint Lab for its support of Scene Cognition Technology Innovation. We also thank all the anonymous reviewers for their constructive comments and suggestions.

6. REFERENCES

[1] Xiaodong Zhang and Houfeng Wang, "A joint model of intent determination and slot filling for spoken language understanding.," in *IJCAI*, 2016.

[2] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *ACL*, 2018.

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

[4] Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," in *ICASSP*, 2021.

[5] Changliang Li, Liang Li, and Ji Qi, "A self-attentive model with gate mechanism for spoken language understanding," in *EMNLP*, 2018.

[6] Leyang Cui and Yue Zhang, "Hierarchically-refined label attention network for sequence labeling," *arXiv preprint arXiv:1908.08676*, 2019.

[7] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *EMNLP/IJCNLP*, 2019.

[8] 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 *ICASSP*, 2021.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[10] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016.

[11] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *EMNLP*, 2016.

[12] Binghui Chen, Weihong Deng, and Jiani Hu, "Mixed high-order attention network for person re-identification," in *ICCV*, 2019.

[13] Min Tan, Fu Yuan, Jun Yu, Guijun Wang, and Xiaoling Gu, "Fine-grained image classification via multi-scale selective hierarchical biquadratic pooling," *ACM TOMM*, 2022.

[14] Jie Jiang and Yi Zhang, "An improved action recognition network with temporal extraction and feature enhancement," *IEEE Access*, 2022.

[15] Jonathan T. Barron, "Continuously differentiable exponential linear units," *CoRR*, vol. abs/1704.07483, 2017.

[16] 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, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018.

[17] Charles T. Hemphill, John J. Godfrey, and George R. Doddington, "The ATIS spoken language systems pilot corpus," in *HLT*, 1990.

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

[19] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[20] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara, "Meshed-memory transformer for image captioning," in *CVPR*, 2020.

[21] Libo Qin, Wanxiang Che, Minheng Ni, Yangming Li, and Ting Liu, "Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding," *IEEE/ACM TASLP*, 2021.

[22] Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *ACL*, 2019.

[23] Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Houfeng Wang, "Graph lstm with context-gated mechanism for spoken language understanding," in *AAAI*, 2020.

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

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