

A CO-INTERACTIVE TRANSFORMER FOR JOINT SLOT FILLING AND INTENT DETECTION

Libo Qin* Tailu Liu* Wanxiang Che† Bingbing Kang Sendong Zhao Ting Liu

Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China

ABSTRACT

Intent detection and slot filling are two main tasks for building a spoken language understanding (SLU) system. The two tasks are closely related and the information of one task can benefit the other. Previous studies either implicitly model the two tasks with multi-task framework or only explicitly consider the single information flow from intent to slot. None of the prior approaches model the bidirectional connection between the two tasks simultaneously in a unified framework. In this paper, we propose a Co-Interactive Transformer which considers the cross-impact between the two tasks. Instead of adopting the self-attention mechanism in vanilla Transformer, we propose a co-interactive module to consider the cross-impact by building a bidirectional connection between the two related tasks, where slot and intent can be able to attend on the corresponding mutual information. The experimental results on two public datasets show that our model achieves the state-of-the-art performance.

Index Terms— Spoken Language Understanding, Intent Detection, Slot Filling, Co-Interactive Transformer

1. INTRODUCTION

Spoken language understanding (SLU) typically consists of two typical subtasks including intent detection and slot filling, which is a critical component in task-oriented dialogue systems [1]. For example, given “*watch action movie*”, intent detection can be seen an classification task to identify an overall intent class label (i.e., *WatchMovie*) and slot filling can be treated as a sequence labeling task to produce a slot label sequence (i.e., *O, B-movie-type, I-movie-type*). Since slots and intent are highly closed, dominant SLU systems in the literature [2, 3, 4, 5, 6] proposed joint model to consider the correlation between the two tasks. Existing joint models can be classified into two main categories. The first strand of work [2, 3] adopted a multi-task framework with a shared encoder to solve the two tasks jointly. While these models outperform the pipeline models via mutual enhancement, they just modeled the relationship implicitly by sharing

parameters. The second strand of work [4, 5, 6] explicitly applied the intent information to guide the slot filling task and achieve the state-of-the-art performance. However, they only considered the single information flow from intent to slot.

We consider addressing the limitation of existing works by proposing a Co-Interactive Transformer for joint slot filling and intent detection. Different from the vanilla Transformer [7], the core component in our framework is a proposed co-interactive module to model the relation between the two tasks, aiming to consider the cross-impact of the two tasks and enhance the two tasks in a mutual way. Specifically, in each co-interactive module, we first apply the label attention mechanism [8] over intent and slot label to capture the initial *explicit intent and slot representations*, which extracts the intent and slot semantic information. Second, the explicit intent and slot representations are fed into a co-interactive attention layer to make mutual interaction. In particular, the *explicit intent representations* are treated as queries and slot representations are considered as keys as well as values to obtain the slot-aware intent representations. Meanwhile, the *explicit slot representations* are used as queries and intent representations are treated as keys as well as values to get the intent-aware slot representations. These above operations can establish the bidirectional connection across intent and slots. The underlying intuition is that slot and intent can be able to attend on the corresponding mutual information with the co-interactive attention mechanism.

The experimental results on two benchmarks SNIPS [9] and ATIS [4] show that our framework achieves significant improvement compared to all baselines. In addition, we incorporate the pre-trained model (BERT) [10] in our framework, which can achieve a new state-of-the-art performance. Code for this paper are publicly available at <https://github.com/kangbrilliant/DCA-Net>.

2. APPROACH

This section describes the details of our framework. As shown in Figure 1, it mainly consists of a shared encoder (§2.1), a co-interactive module (§2.2) that explicitly establishes bidirectional connection between the two tasks, and two separate decoders (§2.3) for intent detection and slot filling.

* Equal contributions.

† Corresponding author.

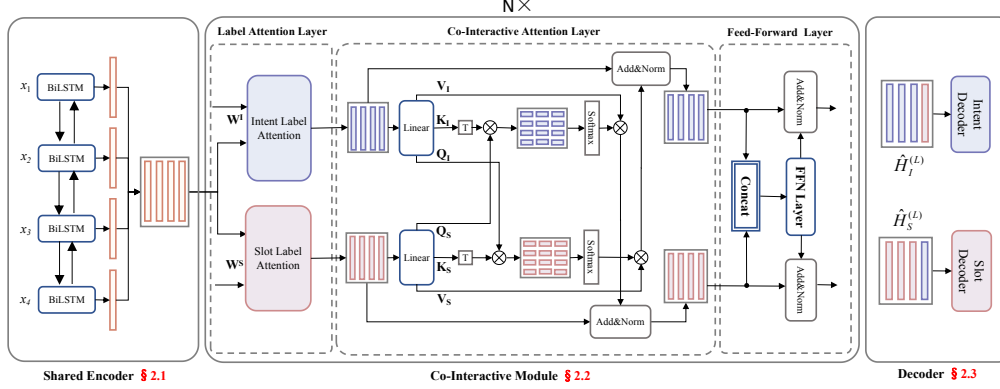


Fig. 1. The illustration of the co-interactive transformer.

2.1. Shared Encoder

We use BiLSTM [11] as the shared encoder, which aims to leverage the advantages of temporal features within word orders. BiLSTM consists of two LSTM layers. For the input sequence $\{x_1, x_2, \dots, x_n\}$ (n is the number of tokens.), BiLSTM reads it forwardly and backwardly to produce a series of context-sensitive hidden states $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ by repeatedly applying the recurrence $\mathbf{h}_i = \text{BiLSTM}(\phi^{emb}(x_i), \mathbf{h}_{i-1})$, where $\phi^{emb}(\cdot)$ represents the embedding function.

2.2. Co-Interactive Module

The Co-Interactive module is the core component of our framework, aiming to build the bidirectional connection between intent detection and slot filling.

In vanilla Transformer, each sublayer consists of a self-attention and a feed-forward network (FFN) layer. In contrast, in our co-interactive module, we first apply a intent and slot label attention layer to obtain the explicit intent and slot representation. Then, we adopt a co-interactive attention layer instead of self-attention to model the mutual interaction explicitly. Finally, we extend the basic FFN for further fusing intent and slot information in an implicit method.

2.2.1. Intent and Slot Label Attention Layer

Inspired by Cui et al. [8] that successfully captures label representations, we perform label attention over intent and slot label to get the explicit intent representation and slot representation. Then, they are fed into co-interactive attention layer to make a mutual interaction directly. In particular, we use the parameters of the fully-connected slot filling decoder layer and intent detection decoder layer as slot embedding matrix $\mathbf{W}^S \in \mathbb{R}^{d \times |\mathcal{S}^{label}|}$ and intent embedding matrix $\mathbf{W}^I \in \mathbb{R}^{d \times |\mathcal{I}^{label}|}$ (d represents the hidden dimension; $|\mathcal{S}^{label}|$ and $|\mathcal{I}^{label}|$ represents the number of slot and intent label, respectively), which can be regarded as the distribution of labels in a certain sense.

Intent and Slot Representations In practice, we use $\mathbf{H} \in \mathbb{R}^{n \times d}$ as the query, $\mathbf{W}^v \in \mathbb{R}^{d \times |\mathcal{V}^{label}|}$ ($v \in \{\mathcal{I} \text{ or } \mathcal{S}\}$) as the key and value to obtain intent representations \mathbf{H}_v with intent label attention:

$$\mathbf{A} = \text{softmax}(\mathbf{H}\mathbf{W}^v), \quad (1)$$

$$\mathbf{H}_v = \mathbf{H} + \mathbf{A}\mathbf{W}^v, \quad (2)$$

where \mathcal{I} denotes the intent and \mathcal{S} represents the slot.

Finally, $\mathbf{H}_I \in \mathbb{R}^{n \times d}$ and $\mathbf{H}_S \in \mathbb{R}^{n \times d}$ are the obtained explicit intent representation and slot representation, which capture the intent and slot semantic information, respectively.

2.2.2. Co-Interactive Attention Layer

\mathbf{H}_S and \mathbf{H}_I are further used in next co-interactive attention layer to model mutual interaction between the two tasks. This makes the slot representation updated with the guidance of associated intent and intent representations updated with the guidance of associated slot, achieving a bidirectional connection with the two tasks.

Intent-Aware Slot and Slot-Aware Intent Representation

Same with the vanilla Transformer, we map the matrix \mathbf{H}_S and \mathbf{H}_I to queries ($\mathbf{Q}_S, \mathbf{Q}_I$), keys ($\mathbf{K}_S, \mathbf{K}_I$) and values ($\mathbf{V}_S, \mathbf{V}_I$) matrices by using different linear projections. To obtain the slot representations to incorporate the corresponding intent information, it is necessary to align slot with its closely related intent. We treat \mathbf{Q}_S as queries, \mathbf{K}_I as keys and \mathbf{V}_I as values. The output is a weighted sum of values:

$$\mathbf{C}_S = \text{softmax}\left(\frac{\mathbf{Q}_S\mathbf{K}_I^\top}{\sqrt{d_k}}\right)\mathbf{V}_I, \quad (3)$$

$$\mathbf{H}'_S = \text{LN}(\mathbf{H}_S + \mathbf{C}_S), \quad (4)$$

where LN represents the layer normalization function [12].

Similarly, we treat \mathbf{Q}_I as queries, \mathbf{K}_S as keys and \mathbf{V}_S as values to obtain the slot-aware intent representation \mathbf{H}'_I . $\mathbf{H}'_S \in \mathbb{R}^{n \times d}$ and $\mathbf{H}'_I \in \mathbb{R}^{n \times d}$ can be considered as leveraging the corresponding slot and intent information, respectively.

2.2.3. Feed-forward Network Layer

In this section, we extend feed-forward network layer to implicitly fuse intent and slot information. We first concatenate \mathbf{H}_I' and \mathbf{H}_S' to combine the slot and intent information.

$$\mathbf{H}_{IS} = \mathbf{H}_I' \oplus \mathbf{H}_S', \quad (5)$$

where $\mathbf{H}_{IS} = (\mathbf{h}_{IS}^1, \mathbf{h}_{IS}^2, \dots, \mathbf{h}_{IS}^n)$ and \oplus is concatenation.

Then, we follow Zhang et al. [3] to use word features for each token, which is formatted as:

$$\mathbf{h}_{(f,t)}^t = \mathbf{h}_{IS}^{t-1} \oplus \mathbf{h}_{IS}^t \oplus \mathbf{h}_{IS}^{t+1}. \quad (6)$$

Finally, FFN layer fuses the intent and slot information:

$$\mathbf{FFN}(\mathbf{H}_{(f,t)}) = \max(0, \mathbf{H}_{(f,t)} \mathbf{W}_1 + b_1) \mathbf{W}_2 + b_2, \quad (7)$$

$$\hat{\mathbf{H}}_I = \text{LN}(\mathbf{H}_I' + \mathbf{FFN}(\mathbf{H}_{(f,t)})), \quad (8)$$

$$\hat{\mathbf{H}}_S = \text{LN}(\mathbf{H}_S' + \mathbf{FFN}(\mathbf{H}_{(f,t)})), \quad (9)$$

where $\mathbf{H}_{(f,t)} = (\mathbf{h}_{(f,t)}^1, \mathbf{h}_{(f,t)}^2, \dots, \mathbf{h}_{(f,t)}^t)$; $\hat{\mathbf{H}}_I$ and $\hat{\mathbf{H}}_S$ is the obtained updated intent and slot information that aligns corresponding slot and intent features, respectively.

2.3. Decoder for Slot Filling and Intent Detection

In order to conduct sufficient interaction between the two tasks, we apply a stacked co-interactive attention network with multiple layers. After stacking L layer, we obtain a final updated slot and intent representations $\hat{\mathbf{H}}_I^{(L)} = (\hat{\mathbf{h}}_{(I,1)}^{(L)}, \hat{\mathbf{h}}_{(I,2)}^{(L)}, \dots, \hat{\mathbf{h}}_{(I,n)}^{(L)})$, $\hat{\mathbf{H}}_S^{(L)} = (\hat{\mathbf{h}}_{(S,1)}^{(L)}, \hat{\mathbf{h}}_{(S,2)}^{(L)}, \dots, \hat{\mathbf{h}}_{(S,n)}^{(L)})$.

Intent Detection We apply *maxpooling* operation [13] on $\hat{\mathbf{H}}_I^{(L)}$ to obtain sentence representation \mathbf{c} , which is used as input for intent detection:

$$\hat{\mathbf{y}}^I = \text{softmax}(\mathbf{W}^I \mathbf{c} + \mathbf{b}_S), \quad (10)$$

$$o^I = \text{argmax}(\hat{\mathbf{y}}^I), \quad (11)$$

where $\hat{\mathbf{y}}^I$ is the output intent distribution; o^I represents the intent label and \mathbf{W}^I are trainable parameters of the model.

Slot Filling We follow E et al. [14] to apply a standard CRF layer to model the dependency between labels, using:

$$\mathbf{O}_S = \mathbf{W}^S \hat{\mathbf{H}}_S^{(L)} + \mathbf{b}_S, \quad (12)$$

$$P(\hat{\mathbf{y}} | \mathbf{O}_S) = \frac{\sum_{i=1} \exp f(y_{i-1}, y_i, \mathbf{O}_S)}{\sum_{y'} \sum_{i=1} \exp f(y'_{i-1}, y'_i, \mathbf{O}_S)}, \quad (13)$$

where $f(y_{i-1}, y_i, \mathbf{O}_S)$ computes the transition score from y_{i-1} to y_i and $\hat{\mathbf{y}}$ represents the predicted label sequence.

3. EXPERIMENTS

3.1. Dataset

We conduct experiments on two benchmark datasets. One is the public ATIS dataset [16] and another is SNIPS [9]. Both

datasets are used in our paper following the same format and partition as in Goo et al. [4] and Qin et al. [6].

In the paper, the hidden units of the shared encoder and the co-interactive module are set as 128. We use 300d GloVe pre-trained vector [17] as the initialization embedding. The number of co-interactive module is 2. L2 regularization used on our model is 1×10^{-6} and the dropout ratio of co-interactive module is set to 0.1. We use Adam [18] to optimize the parameters in our model.

Following Goo et al. [4] and Qin et al. [6], intent detection and slot filling are optimized simultaneously via a joint learning scheme. In addition, we evaluate the performance of slot filling using F1 score, intent prediction using accuracy, the sentence-level semantic frame parsing using overall accuracy.

3.2. Main Results

Table 1 shows the experiment results. We have the following observations: 1) Compared with baselines *Slot-Gated* and *Stack-Propagation* that only leverage intent information to guide the slot filling, our framework gain a large improvement. The reason is that our framework consider the cross-impact between the two tasks where the slot information can be used for improving intent detection. It's worth noticing that the parameters between our model and *Stack-Propagation* is of the same magnitude, which further verifies that contribution of our model comes from the bi-directional interaction rather than parameters factor. 2) *SF-ID Network* and *CM-Net* also can be seen as considering the mutual interaction between the two tasks. Nevertheless, their models cannot model the cross-impact simultaneously, which limits their performance. Our framework outperforms *CM-Net* by 6.2% and 2.1% on overall acc on SNIPS and ATIS dataset, respectively. We think the reason is that our framework achieves the bidirectional connection simultaneously in a unified network. 3) *Our framework + BERT* outperforms the *Stack-Propagation + BERT*, which verifies the effectiveness of our proposed model whether it's based on BERT or not.

3.3. Analysis

Impact of Explicit Representations We remove the intent attention layer and replace \mathbf{H}_I with \mathbf{H} . This means that we only get the slot representation explicitly, without the intent semantic information. We name it as *without intent attention layer*. Similarly, we perform the *without slot attention layer* experiment. The result is shown in Table 2, we observe that the slot filling and intent detection performance drops, which demonstrates the initial explicit intent and slot representations are critical to the co-interactive layer between the two tasks.

Co-Interactive Attention vs. Self-Attention Mechanism

We use the self-attention layer in the vanilla Transformer instead of the co-interactive layer in our framework, which can be seen as no explicit interaction between the two tasks.

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Slot-Gated Atten [4]	88.8	97.0	75.5	94.8	93.6	82.2
SF-ID Network [14]	90.5	97.0	78.4	95.6	96.6	86.0
CM-Net [15]	93.4	98.0	84.1	95.6	96.1	85.3
Stack-Propagation [6]	94.2	98.0	86.9	95.9	96.9	86.5
Our framework	95.9	98.8	90.3	95.9	97.7	87.4
Stack-Propagation + BERT [6]	97.0	99.0	92.9	96.1	97.5	88.6
Our framework + BERT	97.1	98.8	93.1	96.1	98.0	88.8

Table 1. Slot filling and intent detection results on two datasets.

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
without intent attention layer	95.8	98.5	90.1	95.6	97.4	86.6
without slot attention layer	95.8	98.3	89.4	95.5	97.6	86.7
self-attention mechanism	95.1	98.3	88.4	95.4	96.6	86.1
with intent-to-slot	95.6	98.4	89.3	95.8	97.1	87.2
with slot-to-intent	95.4	98.7	89.4	95.5	97.7	87.0
Our framework	95.9	98.8	90.3	95.9	97.7	87.4

Table 2. Ablation experiments on the SNIPS and ATIS datasets.

Specifically, we concatenate the \mathbf{H}_S and \mathbf{H}_I output from the label attention layer as input, which is fed into the self-attention module. The results are shown in Table 2, we observe that our framework outperforms the *self-attention mechanism*. The reason is that *self-attention mechanism* only model the interaction implicitly while our co-interactive layer can explicitly consider the cross-impact between two tasks.

Bidirectional Connection vs. One Direction Connection

We only keep one direction of information flow from intent to slot or slot to intent. We achieve this by only using one type of information representation as queries to attend another information representations. We name it as *with intent-to-slot* and *with slot-to-intent*. From the results in Table 2. We observe that our framework outperforms *with intent-to-slot* and *with slot-to-intent*. We attribute it to the reason that modeling the mutual interaction between slot filling and intent detection can enhance the two tasks in a mutual way. In contrast, their models only consider the interaction from single direction of information flow.

4. RELATED WORK

Different classification methods, such as support vector machine (SVM) and RNN [19, 20], have been proposed to solve Intent detection. Meanwhile, the popular methods are conditional random fields (CRF) [21] and recurrent neural networks (RNN) [22, 23] are proposed to solve slot filling task.

Recently, many dominant joint models [2, 3, 4, 5, 24, 6, 25] are proposed to consider the closely correlated relationship between two correlated tasks. The above studies either adopt a multi-task framework to model the relationship between slots and intent implicitly or leverages intent information to guide slot filling tasks explicitly. Compared with their models, we propose a co-interactive transformer framework, which simultaneously considers the cross-impact and estab-

lish a directional connection between the two tasks while they only consider the single direction information flow or implicitly model the relationship into a set of shared parameters.

Meanwhile, Wang et al. [26], E et al. [14], and Liu et al. [15] propose models to promote slot filling and intent detection via mutual interaction. Compared with their methods, the main differences are as following: 1) E et al. [14] introduce a SF-ID network, which includes two sub-networks iteratively achieve the flow of information between intent and slot. Compared with their models, our framework build a bidirectional connection between the two tasks simultaneously in a unified framework while their frameworks must consider the iterative task order. 2) Liu et al. [15] propose a collaborative memory block to implicitly consider the mutual interaction between the two tasks, which limits their performance. In contrast, our model proposes a co-interactive attention module to explicitly establish the bidirectional connection in a unified framework.

5. CONCLUSION

In our paper, we proposed a co-interactive transformer for joint model slot filling and intent detection, which enables to fully take the advantage of the mutual interaction knowledge. Experiments on two datasets show the effectiveness of the proposed models and our framework achieves the state-of-the-art performance.

6. ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 61772153. This work was supported by the Zhejiang Lab’s International Talent Fund for Young Professionals.

7. REFERENCES

- [1] Gokhan Tur and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.
- [2] Bing Liu and Ian Lane, “Attention-based recurrent neural network models for joint intent detection and slot filling,” *arXiv preprint arXiv:1609.01454*, 2016.
- [3] Xiaodong Zhang and Houfeng Wang, “A joint model of intent determination and slot filling for spoken language understanding.” in *Proc. of IJCAI*, 2016.
- [4] 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 *Proc. of NAACL*, 2018.
- [5] Changliang Li, Liang Li, and Ji Qi, “A self-attentive model with gate mechanism for spoken language understanding,” in *Proc. of EMNLP*, 2018.
- [6] 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 *Proc. of EMNLP*, Nov. 2019.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *NIPS*. 2017.
- [8] Leyang Cui and Yue Zhang, “Hierarchically-refined label attention network for sequence labeling,” in *Proc. of EMNLP*, 2019.
- [9] 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.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [11] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, 1997.
- [12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, “Layer normalization,” 2016.
- [13] Yoon Kim, “Convolutional neural networks for sentence classification,” in *Proc. of EMNLP*, Oct. 2014.
- [14] Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song, “A novel bi-directional interrelated model for joint intent detection and slot filling,” in *Proc. of ACL*, 2019.
- [15] 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*, 2019.
- [16] 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.
- [17] Jeffrey Pennington, Richard Socher, and Christopher Manning, “Glove: Global vectors for word representation,” in *Proc. of EMNLP*, 2014.
- [18] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [19] Patrick Haffner, Gokhan Tur, and Jerry H Wright, “Optimizing svms for complex call classification,” in *In Proc. of ICASSP*, 2003.
- [20] Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran, “Deep belief nets for natural language call-routing,” in *Proc. of ICASSP*, 2011.
- [21] Christian Raymond and Giuseppe Riccardi, “Generative and discriminative algorithms for spoken language understanding,” in *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [22] Puyang Xu and Ruhi Sarikaya, “Convolutional neural network based triangular crf for joint intent detection and slot filling,” in *Proc. of ASRU*, 2013.
- [23] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, “Spoken language understanding using long short-term memory neural networks,” in *SLT*, 2014.
- [24] Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang, “A neural multi-task learning framework to jointly model medical named entity recognition and normalization,” in *Proc. of AAAI*, 2019.
- [25] Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev, “Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering,” in *Proc. of ACL*, July 2018.
- [26] 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*, 2018.