Using Bidirectional Transformer-CRF for Spoken Language Understanding

Linhao Zhang and Houfeng Wang(✉)

MOE Key Lab of Computational Linguistics, Peking University,
Beijing 100871, China
{zhanglinhao,wanghf}@pku.edu.cn

Abstract. Spoken Language Understanding (SLU) is a critical component in spoken dialogue systems. It is typically composed of two tasks: intent detection (ID) and slot filling (SF). Currently, most effective models carry out these two tasks jointly and often result in better performance than separate models. However, these models usually fail to model the interaction between intent and slots and ties these two tasks only by a joint loss function. In this paper, we propose a new model based on bidirectional Transformer and introduce a padding method, enabling intent and slots to interact with each other in an effective way. A CRF layer is further added to achieve global optimization. We conduct our experiments on benchmark ATIS and Snips datasets, and results show that our model achieves state-of-the-art on both tasks.

Keywords: SLU · Transformer · CRF · Joint Method.

1 Introduction

Spoken language understanding (SLU) is an important part of a dialogue system. An utterance of a user is often first transcribed to text by an automatic speech recognizer (ASR) and then converted by the SLU component to the structured representations. The result of SLU is passed to dialogue management module to update dialogue state and make dialogue policy. Therefore, the performance of SLU is critical to building an effective dialogue system [24].

SLU usually involves intent detection (ID) and slot filling (SF). Typically, ID is regarded as a semantic utterance classification problem and different classification methods can be applied [3, 6]. Meanwhile, SF is usually treated as a sequence labeling problem. that maps a word sequence $\mathbf{x} = (x_1, ..., x_T)$ to the corresponding slot label sequence $\mathbf{y} = (y_1, ..., y_T)$. Popular approaches to perform SF include conditional random fields (CRFs) [13], support vector machines (SVMs) [12] and maximum entropy Markov models (MEMM) [16].

In recent years, neural network approaches have demonstrated outstanding performance in a variety of NLP tasks, and RNN-based methods have been widely applied in the SLU area. [17, 5]. Despite the success they have achieved, the sequential nature of RNNs precludes any parallelization. Besides, in SLU,

Utterance	show	flights	from	Seattle	to	San	Diego	tomorrow
Slots	O	O	O	B-fromloc	O	B-toloc	I-toloc	B-departdate
Intent	Flight							

Fig. 1. An example of ATIS sentence with annotated slots using the IOB scheme and intent. The B- prefix before a tag indicates that the tag is the beginning of a chunk, and an I- prefix before a tag indicates that the tag is inside a chunk. An O tag indicates that a token belongs to no chunk.

slots are determined not only by the associated items, but also by context. As shown in Figure 1, the corresponding slot label for city name *Seattle* is *B-fromloc*, but it could also be *B-toloc*, if the utterance is *show flights from San Diego to Seattle tomorrow*. Note that this is different from Named Entity Recognition (NER), which in general has less dependency on context than SF task (in the above example, *Seattle* can be simply recognized as a *Location*). Compared to RNNs, we believe that Transformer, which is based on self-attention mechanism and capable of learning the internal structure of a sentence [20], is better at capturing such dependency. Besides, it allows for more parallelization within the sentences.

CRF has long been known to be able to explicitly model the dependency among the output labels, which is a very advantageous feature for sequence labeling task [15]. It has been widely used in sequence labeling tasks like named entity recognition and Chinese word segmentation [10]. In SLU areas, it has also been exploited [21, 25]. In this work, we add a CRF layer for SF to achieve global optimization.

ID and SF are traditionally treated separately. In recent years, joint models have been proposed and lead to better performance [14, 7]. The main rationale of such methods is that these two tasks are not independent but intrinsically linked. For example, an utterance is more likely to contain departure and arrival cities if its intent is to find a flight, and vice versa [25]. To perform these two tasks jointly, we first pad the input sequence with a special token *BOS* at the beginning and use the representation of this token learned by bidirectional Transformer to predict the intent of the whole sentence. We argue that this method is especially suitable for joint ID and SF due to its ability to allow intent and slots to directly attend to each other. Previous work links these two tasks only by a joint loss function, thus may fail to make full use of the interaction between these two tasks. [5] tackles this problem via slot-gated mechanism, leveraging intent vector to influence slots prediction. However, this kind of influence is only one way.

Our contributions are three-fold:

1)We analyze and highlight the advantageous features of bidirectional Transformer when applied to SLU. To the best of our knowledge, this is the first attempt to introduce the Transformer architecture into this area.

2)We propose a padding method that allows slots and intent to interact with each other in an elegant and effective way.

3)Experiments demonstrate that our new model achieves state-of-the-art for both ID and SF. Specifically, on ATIS, our model achieves 97.2% accuracy on ID and 95.1% F1 score on SF. On snips, the performance boost is more significant, with ID accuracy of 98.9% and SF F1 score of 93.3 %.

The rest of the paper is organized as follows. In Section 2 we introduce our proposed model. We give our experiment settings and results in section 3. The related work is surveyed in Section 4. The conclusion is given in the last section.

2 Model

Figure 2 gives an overview of our proposed model. The input is a sequence of words in an utterance, and the output is the annotated slots using IOB scheme, plus the intent of the whole utterance. A detailed description is given below.

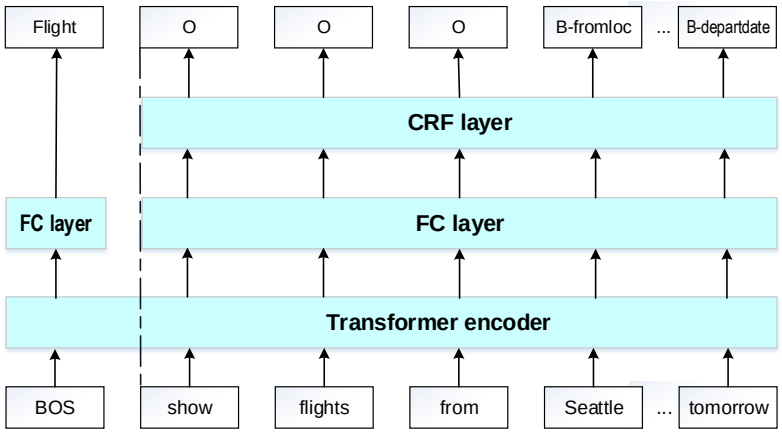

Fig. 2. The architecture of the proposed model. We pad the input utterance with a BOS symbol, and use the representation of this symbol to perform ID. For SF, we utilize a CRF layer to perform global optimization.

2.1 Word Representations

We first convert the input word sequence $(w_1, ..., w_T)$ to a sequence of word embeddings $(\hat{e_1}, ..., \hat{e_T})$ and use these embeddings as model input.

Given the limited size of the ATIS dataset, one may assume that using pre-trained word embeddings to initialize the embedding layer may lead to better performance. We examine this idea with *GloVe* vectors [18], and did not notice any improvement in performance. We instead employ a simple randomly initialized embedding layer.

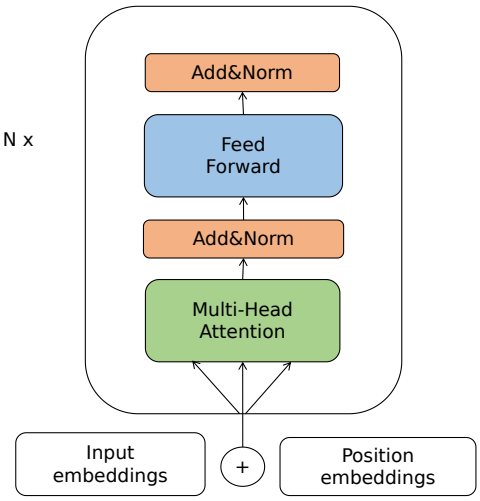

Fig. 3. The structure of Transformer encoder [20], which is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a fully connected feed-forward network. Residual connection [8] and layer normalization [2] are employed around each of the two sub-layers

Following [17], we use a context word window as the input to our model. Given d the window size (which is a hyperparameter), we define the d-context window as the ordered concatenation of $2d+1$ word embeddings,i.e. d previous word embeddings followed by the word of interest and next d word embeddings. Formally,

$$e_t = [\hat{e_{t-d}}, \ldots, \hat{e_t}, \ldots, \hat{e_{t+d}}] \tag{1}$$

Our model input is these concatenated word embeddings $(e_1, ..., e_T)$. In this window approach, one might wonder how to build a d-context window for the first/last words of the sentence. We adopt a simple approach to replicate their word embeddings several times, depending on the exact positions of the words and the window size d, and then perform the concatenation.

2.2 Transformer and Self-Attention Mechanism

The Transformer was first proposed in [20] for the task of Neural Machine Translation (NMT). It consists of a bidirectional Transformer ("Transformer encoder") and a left-to-right Transformer ("Transformer decoder"). The encoder first maps an input of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $(z_1, ..., z_n)$, which is later used by the decoder to generate an output sequence $(y_1, ..., y_n)$ of symbols one at a time. Note that in our model only the Transformer encoder is employed.

As shown in Figure 2 The Transformer encoder is composed of N identical layers, each of which consists of two sub-layers, namely the self-attention layer and the position-wise fully connected layer. The core idea behind the Transformer encoder is the self-attention mechanism, which relates different positions of a sentence in order to compute a representation of it. Given Q, K, V the packed queries, keys and values respectively and d_k the dimension of keys, The attention mechanism used in Transformer can be formally put as:

$$Att(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{2}$$

$$V_{att} = Att(Q, K)V \tag{3}$$

[20] find that multi-head attention perform better than a single attention function. The intuition behind is that if we only computed a single attention weighted sum of the values, capturing different aspects of the input would be difficult. To learn diverse representations, the multi-head attention applies different linear transformations to the values, keys, and queries for each "head" of attention. Following their approach, we first project the queries, keys and values h times with different linear projections to d_k, d_k and d_v dimensions respectively. We then perform the attention function on each of these projected vectors, resulting in d_v-dimensional output values, which are concatenated and once again projected, yielding the final values. Formally,

$$Multi(Q, K, V) = [head_1, \ldots, head_{\mathrm{h}}]W^O \tag{4}$$

$$head_i = Attn\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{5}$$

where the projections are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$ and $W^O \in R^{hd_v \times d_{model}}$.

While in a self-attention layer, Q, K and V are from the same place, namely the previous layer in the encoder. Thus each position in the encoder can attend to all positions in the previous layer. This feature allows the Transformer to ignore the distance between words and directly compute dependency relationships, making it especially suitable for tasks like SF that depends heavily on context.

Apart from attention sub-layers, each of the layers in the Transformer encoder contains a fully connected feed-forward network, which consists of two linear transformations with a ReLU activation in between (see Figure 3).

2.3 Padding Method for Joint ID and SF

To carry out SF and ID jointly, we propose a simple yet highly effective method. We first pad the input sequence with a special token *BOS* at the beginning and use the representation of this token learned by bidirectional Transformer to predict the intent of the whole utterance. The new input and output sequences for our model are:

$$X = BOS, x_1, ..., x_n \tag{6}$$

$$Y = intent, y_1, ..., y_n \tag{7}$$

In Subsection 2.2, we mention that the self-attention mechanism of Transformer allows each position in the encoder to attend to all positions in the previous layer. Combined with our padding method, intent and slots can now directly interact with each other. Most of the previous joint models model the relationship between intent and slots implicitly by a joint loss function [25, 14, 7], or, by mechanisms like "slot-gated" [5], leveraging intent vector to influence slots prediction (note that this kind of influence is one way). Our model, on the other hand, allows intent and slots to influence each other in both directions while maintains simplicity. By considering the cross-impact between intent and slots, both ID and SF get improved.

2.4 Task Specific Layers

Given d_m the dimension of bidirectional Transformer and l the length of input sentence (including the padding in the beginning), the output of bidirectional Transformer is a matrix $T \in R^{l*d_m}$. To perform ID, we extract the first row of T (named as T^0), and apply the softmax function to the linear transformation of T^0 to get the probability distribution y^i over all intent labels:

$$y^i = softmax(W^i T^0 + b^i) \tag{8}$$

where W_i and b_i are model parameters.

The remaining part of T (named as $T^- \in R^{(l-1)*d_m}$) is used for SF, with each row corresponding to a position to be labeled ($l - 1$ in total). We then perform a linear transformation:

$$S^s = W^s T^- + B^s \tag{9}$$

where W_s and b_s are model parameters.

Similar to Equation 8, we can then directly apply a softmax function in order to get the final probability distribution over all the slot labels. However, this method has the disadvantage of allowing illegal label combinations to be outputted. For example, an *I-fromloc* after a *B-toloc* is clearly invalid and yet could be potentially created by such a method.

To address this problem, we feed S^s into a CRF layer, which can add some constraints to the final predicted labels to ensure that they are valid. These constraints are learned by the CRF layer automatically from the training data.

The loss function of the model is the sum of negative log-probability of the correct tag sequence for both intent and slot.

$$\mathcal{L}(\theta) = \Sigma_{(l^s, l^i, U) \in \mathcal{D}} \left(\alpha \mathcal{L}^s(\theta) + \mathcal{L}^u(\theta) \right) \tag{10}$$

where D is the dataset. $\mathcal{L}^u(\theta)$ and $\mathcal{L}^s(\theta)$ are loss for ID and SF respectively. We use a weighted factor α to adjust the importance of the two tasks.

Table 1. Statistics of the ATIS and Snips dataset.

	ATIS	**Snips**
#Slots	120	72
#Intents	21	7
Vocabulary Size	722	11241
Training Set	4478	13084
Dev Set	500	700
Test Set	893	700

Table 2. Intents and examples of the Snips dataset.

Intent	**Utterance Example**
SearchCreativeWork	Find me the I, Robot television show
GetWeather	Is it windy in Boston, MA right now?
BookRestaurant	I want to book a highly rated restaurant tomorrow night
PlayMusic	Play the last track from Beyonc off Spotify
AddToPlaylist	Add Diamonds to my roadtrip playlist
RateBook	Give 6 stars to Of Mice and Men
SearchScreeningEvent	Check the showtimes for Wonder Woman in Paris

3 Experiment

3.1 Datasets

To fully evaluate the proposed model, we conducted experiments on two datasets: ATIS and Snips. The details of these two dataset are given below:

ATIS The Airline Travel Information Systems (ATIS) [9] dataset has long been exploited in SLU. There are some variants of the ATIS dataset. In this work, we use the same one as used in [17, 25, 14]. There are 4,978 utterances in the training set and 893 in the test set. There are in total 127 distinct slot labels and 17 different intent types.

The ATIS dataset also has extra named entity (NE) features marked via table lookup, which are utilized by many of the previous researchers [4, 17, 25]. For the sake of generalization, we did not utilize these features in our study.

Snips We obtain this dataset from [5]. It is in the domain of personal assistant commands. Compared to the ATIS corpus, the Snips dataset is more complicated in terms of vocabulary size and the diversity of intent and slots. There are 13,084 utterances in the training set and 700 utterances in the test set, with a development set of 700 utterances. There are 72 slot labels and 7 intent types. As shown in Table 2, the diversity of intents and slots is an important feature of Snips dataset. Slots of places in ATIS are generally limited to American cities and intents are all about flight information, while Snips contains intents like *RateBook* and *GetWeather* that come from total different topics.

Table 3. Intent accuracy and slot filling F1 scores on ATIS and Snips datasets(%). The reported results are from [5].

Model	ATIS		Snips	
	ID	SF	ID	SF
Bi-LSTM [7]	92.6	94.3	96.9	87.3
Attention-Based RNN [14]	91.1	94.2	96.7	87.8
Slot-Gated(Full Attn.) [5]	93.6	94.8	97.0	88.8
Slot-Gated(Intent Attn.) [5]	94.1	**95.2**	96.8	88.3
Our model	**97.2**	95.1	**98.9**	**93.3**

Table 4. Comparison between joint and separate models on the Snips dataset. Joint model is our proposed model in Figure 2. Separate-ID model only contains the shared layer and ID specific layer, and it is the same way for Separate-SF model.

Model	ID	SF
Separate-ID	96.4	-
Separate-SF	-	92.8
Joint	**98.9**	**93.3**

3.2 Training Procedure

We trained our models on a single NVIDIA GeForce GTX 1080 GPU. The dimension of word embedding is set to 80 and 120 for ATIS and Snips dataset respectively. The context window size is 1 for both datasets. Dropout layers are applied on both input and output vectors during training for regularization; the dropout rate is set to 0.5. The number of layers of bidirectional Transformer is set to 6. The batch size is set to 32. We use Adam optimizer for the training process. All these hyperparameters are chosen using the validation set.

We use Adam [11] for the training process to minimize the cross-entropy loss, with learning rate $= 10^{-3}$ $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The CRF layer is implemented with AllenNLP, which is an open-source NLP research library built on PyTorch.

3.3 Experimental Results

Overall Performance We use F1 score and accuracy as evaluation metrics for SF and ID respectively. Note that some utterances in ATIS corpus have more than one intent labels. Following [5], we require that all of these intent labels have to be correctly predicted if a sentence is counted as a correct classification.

We compare our model against some baselines and the results are demonstrated in Table 3. We here use the scores reported in [5] since we use the same dataset and evaluation settings.

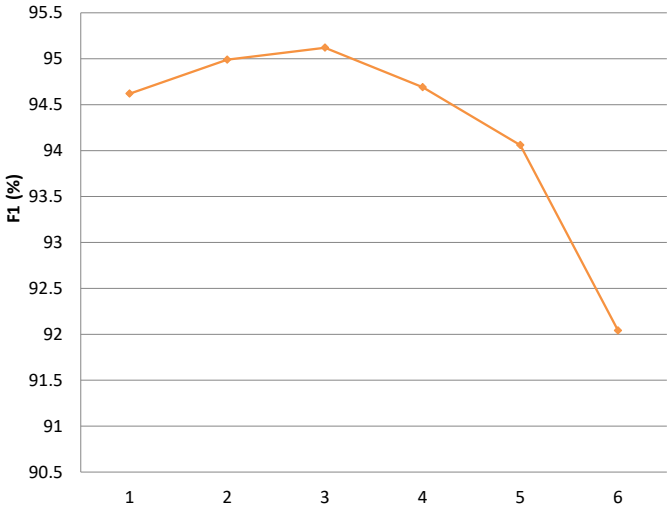

Fig. 4. Influence of the number of layers of Transformer encoder on F1 score.

On the ATIS dataset, we achieve the state-of-the-art for SF and outperform the best reported results for ID by a large margin. On the Snips dataset, the performance boost is more significant, with 1.9% and 4.5% absolute improvement for ID and SF respectively. We contribute the improvement to the following reasons: 1)All but a few previous works are RNN-based, and Transformer has recently shown its superior fitting ability in many other NLP areas. 2)Our padding method allows intent and slots to interact with each other in a simple and effective way.

Generally speaking, our model performs better on the Snips dataset, which is larger and more diverse. This difference shows the potential for our model to be applied in an open domain area.

Joint vs Separate To further assess the effectiveness of our padding method, we compare our joint model with separate models on the Snips dataset, and the results are shown in Table 4. Apparently, the joint model outperforms the separate models on both tasks (0.5% and 2.5% absolute improvement for SF and ID respectively). The results suggest that the correlation between slots and intent is learned by our joint model and contributes to both tasks.

Layers of Transformer In the experiments, we also notice that the layer of Transformer influences the final performance a lot. As shown in Figure 4, we achieve the best F1 score with 3 layers of Transformer. When the number of layers grows larger than 3, the performance drops significantly. We also notice that simple concatenation of representations from different layer can lead to better performance, and we leave this to future work.

4 Related Work

Historically, SF task originated mostly from non-commercial projects such as the ATIS project, on the other hand, ID emerged from the call classification systems after the success of the early commercial interactive voice response (IVR) applications used in call centers. Many traditional machine learning approaches have since been used in this area [6, 19, 12] .

In recent years, RNN-based methods have defined the state-of-the-art in SLU research. [23] adapted RNN language models to perform SLU, outperforming previous CRF result by a large margin. They attribute the superior performance to the task-specific word representations learned by the RNN. [17] investigated different kinds of RNNs for slot filling and shown that Elman RNN performed better than Jordan RNN. [22] used a deep LSTM architecture and investigated the relative importance of each gate in the LSTM by setting other gates to a constant and only learning particular gates.

There have been many attempts to learn ID and SF jointly. [21] first proposed a joint model for ID and SF based on convolutional neural network (CNN). [14] proposed an attention-based neural network model and beat the state-of-the-art on both tasks. [25] used a GRU-based model and max-pooling method to jointly learn these two tasks. [7] proposed a multi-domain, multi-task sequence tagging approach. Despite their success, these models did not explicitly model the interaction between ID and SF and only tied these two tasks through a joint loss function. [5] pointed this problem out and tackled this with gated mechanism, leveraging intent vector to influence slots prediction. [1] extended this idea by combining the intent vector with self-attention representations.

Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence, it is especially efficient at learning long-range dependencies. Based on the self-attention mechanism. The Transformer was first proposed in [20] for NMT and achieved huge improvement on BLEU scores. Some researches have successfully adapted this architecture to other tasks like sentence simplification [26] and video captioning [27]. However, to the best of our knowledge, this architecture has not been applied in the SLU area.

5 Conclusion

Most previous works for ID and SF are RNN-based. In this paper, we analyze and highlight the advantageous features of bidirectional Transformer when applied to SLU. To our knowledge, this is the first attempt to introduce the Transformer architecture into this area. Using a simple padding method, we jointly perform SF and ID and boost the performance for both tasks. Experiments show that our model outperforms the state-of-the-art results by a large margin. We encourage more researches in this direction.

6 Acknowledgments

Our work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002101 and National Natural Science Foundation of China under GrantNo.61433015.

References

1. A Self-Attentive Model with Gate Mechanism for Spoken Language Understanding. Emnlp pp. 3824–3833 (2018)
2. Ba, J., Kiros, R., Hinton, G.E.: Layer normalization. CoRR (2016)
3. Deng, L., Tur, G., He, X., Hakkani-Tur, D.: Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In: Spoken Language Technology Workshop (SLT), 2012 IEEE. pp. 210–215. IEEE (2012)
4. Deoras, A., Sarikaya, R.: Deep belief network based semantic taggers for spoken language understanding. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH pp. 2713–2717 (01 2013)
5. Goo, C.W., Gao, G., Hsu, Y.K., Huo, C.L., Chen, T.C., Hsu, K.W., Chen, Y.N.: 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). vol. 2, pp. 753–757 (2018)
6. Haffner, P., Tur, G., Wright, J.H.: Optimizing svms for complex call classification. In: Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. vol. 1, pp. I–I. IEEE (2003)
7. Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.N., Gao, J., Deng, L., Wang, Y.Y.: Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In: Interspeech. pp. 715–719 (2016)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
9. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: 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)
10. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. CoRR (2015)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR (2015)
12. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. pp. 1–8. Association for Computational Linguistics (2001)
13. Lafferty, J., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning **8**(June), 282–289 (2001)
14. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv preprint arXiv:1609.01454 (2016)
15. Ma, S., Sun, X.: A new recurrent neural crf for learning non-linear edge features. arXiv preprint arXiv:1611.04233 (2016)

16. McCallum, A., Freitag, D., Pereira, F.: Maximum entropy markov models for information extraction and segmentation. In: ICML (2000)
17. Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., et al.: Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**(3), 530–539 (2015)
18. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
19. Tür, G., Hakkani-Tür, D.Z., Heck, L.P., Parthasarathy, S.: Sentence simplification for spoken language understanding. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 5628–5631 (2011)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
21. Xu, P., Sarikaya, R.: Convolutional Neural Network Based Triangular Crf for Joint Intent Detection and Slot Filling pp. 78–83 (2013)
22. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: Spoken Language Technology Workshop (SLT), 2014 IEEE. pp. 189–194. IEEE (2014)
23. Yao, K., Zweig, G., Hwang, M.Y., Shi, Y., Yu, D.: Recurrent neural networks for language understanding. In: Interspeech. pp. 2524–2528 (2013)
24. Zhang, X., Ma, D., Wang, H.: Learning dialogue history for spoken language understanding. In: NLPCC (2018)
25. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: IJCAI. pp. 2993–2999 (2016)
26. Zhao, S., Meng, R., He, D., Andi, S., Bambang, P.: Integrating Transformer and Paraphrase Rules for Sentence Simplification. arXiv preprint arXiv:1810.11193 (2018)
27. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8739–8748 (2018)