Attention-Based RNN Model for Joint Extraction of Intent and Word Slot Based on a Tagging Strategy

Dongjie Zhang[1,2], Zheng Fang[1,2], Yanan Cao[2(✉)], Yanbing Liu[2], Xiaojun Chen[2], and Jianlong Tan[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhangdongjie, fangzheng, caoyanan, liuyanbing, chenxiaojun, tanjianlong}@iie.ac.cn

Abstract. In this paper, we proposed an attention-based recurrent neural network model based on a tagging strategy for intent detection and word slot extraction. Unlike other joint models dividing the joint task into two sub-models by sharing parameters, we explore a tagging strategy to incorporate the intent detection task and word slot extraction task in a sequence labeling model. We implemented experiments on a public dataset and the results show that the tagging strategy methods outperform most of the existing pipelined and joint methods. Our tagging strategy model obtained 97.65% accuracy rate on intent detection task and 95.15% F1 score on word slot extraction task.

Keywords: Intent detection · Word slot extraction · Joint model
Attention mechanism · Tagging strategy

1 Introduction

Intent detection and word slot extraction are two basic issues in the field of Natural Language Understanding and these two tasks are usually handled separately [19]. Intent detection and word slot extraction can be regarded as a sentence classification and sequence tagging task respectively. Traditionally, we solve these problems in a sequential order, extracting the word slots first and then detecting the intent of the given sentence. This separated framework makes the task easy to handle and can deal with different subtask issues more flexibly. It is assumed that these two tasks have no correlation between them which enables them to be treated as an independent model, however, in many cases this is not true. Thus, the results of the word slot extraction can affect the outcome of the intent detection by the propagation of errors.

Compared with the pipeline models, the joint learning framework handles the two tasks using a single model [2]. The joint model can integrate the information of word slots and of intent by sharing collective parameters and it has been shown to perform well on the joint extraction task [20]. These joint models can make the intent detection

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11141, pp. 178–188, 2018.
https://doi.org/10.1007/978-3-030-01424-7_18

and word slot extraction process simpler as we only need to train one model to fine-tune the tasks.

Although the aforementioned joint methods can handle the two subtasks in a single model, they can also produce redundant information by extracting word slots and intents separately. Generally, these frameworks need two classifiers which have separate label collections: one for intent extraction and another for word slot labeling. So the total number of labels is the combined size of the two label collections. However, this may produce redundant labeling results, that is, if there is a slot *s* that never appears in the intent *i*, the model may still give the result of labeling a word as word slot *s* along with the intention *i*. In addition, it's inevitable to propagate the error of two classifiers to each other during training the joint model. In our work, we model the relation of word slots and intent directly by using only one sequential label classifier instead of extracting the word slots and intents separately. We redefined a set of tags containing the information of word slot and the intent of the whole sentence. Based on this tagging strategy, the joint extraction of word slot and intent can be converted into a sequence tagging problem. With this strategy, we can easily use sequence-to-sequence models to handle the two tasks simultaneously without complicated feature engineering.

However, one word slot may have various intents in different sentences and the words indicating the intent of the sentence may locate far away from the current input word. Many sequence labeling model are capable of capturing long-distance dependence information but they still strongly focus on the parts around the current input word. The attention mechanism which has made satisfactory effect in the field of machine translation [1] can effectively learn global attention information of the sequence by emphasizing the influence of key words on the model results. Specially, we wonder if the attention mechanism can be utilized in and improve our joint tagging model. So we implemented the attention mechanism on our joint model to make it more sensitive to key information, especially the long-distance information indicating the intent.

In this paper, we focus on resolving the issue of redundant labeling results, propagation of interactions intrinsically in the pipeline as well as the traditional joint training models on word slot extraction task and intent detection task. Based on the motivation, we applied a tagging strategy accompanied with an end-to-end model to settle the problem by transforming the joint extraction task into a sequence tagging problem. In order to solve the influence of the diversified relation between word slots and intentions, we introduce global sentence information through attention mechanism to enhance the effect of sentence intent. Experiments on ATIS data set show that our joint model significantly outperforms pipeline and traditional joint models.

The rest of our paper is organized as follows. In Sect. 2, we introduce the related works of RNN sequence labeling model and the attention mechanism for sequence labeling. In Sect. 3, we describe our labeling strategy and end-to-end RNN extraction models in detail. In Sect. 4 we mainly show the settings and results of our experiments. Finally, we conclude the work in Sect. 5.

2 Related Works

Intent detection and word slot extraction are corresponded to two fundamental problems–text classification and sequence labeling, which are the basis of many natural language applications and are usually solved in a pipeline manner. For Intent detection, Support Vector machines (SVMs) [3], deep neural network methods [14] and Convolutional Neural Networks (CNNs) [7] have been widely used. The boosting method [16] and its improved method with dependency parsing-based sentence simplification [17] can handle the complex, longer and natural utterances more effectively. The adaptation of the recursive neural network also achieved competitive performance on the intent detection task [2]. In case of word slot extraction, few of the most popular methods include Maximum Entropy Markov Models (MEMMs) [10], Conditional Random Fields (CRFs) [13] and Recurrent Neural Networks (RNNs) [11]. Label dependency is beneficial for word slot extraction task by feeding previous output label [9]. The RNN-CRF networks can also be used in word slot extraction task [5]. In general, simple Recurrent Neural Networks and Convolutional Neural Networks have shown to significantly outperform the previous state-of-the-art Maximum Entropy Markov Models and Conditional Random Fields and the deep Long Short-Term Memories (LSTMs) was emphatically proposed to be applied to the word slot extraction task [20]. In addition, the joint training model has become a research hotspot. The joint model of Recursive Neural Networks integrated two subtasks into one compositional model by providing an elegant mechanism for incorporating both discrete syntactic structure and continuous-space word and phrase representations [2]. The CNN-CRF model can be jointly trained by extracting features automatically from CNN layers and sharing with the intent model [19].

Recently, a novel tagging strategy has been proposed in joint extraction of entities and relations [22]. Results show that the tagging methods are better than most of the existing pipelined and joint learning methods without identifying entities and relations separately. This task mainly focuses on extracting a triplet consisting of two entities and the relation of the two entities. Unlike traditional models, this work proposed a tagging strategy that label triples directly rather than extracting entities and relationships separately. To implement this tagging strategy, a new set of labels containing information about the entities and the relation between them has been designed. With this tagging strategy, the joint extraction of entities and relations can be transformed into a sequence labeling problem. In this way, the sequence labeling model can be conveniently used to handle the joint task without complex feature engineering. However, this tagging strategy still has deficiencies in identifying overlapping relationships and the diversity association between two corresponding entities still needs to be refined.

3 Proposed Methods

3.1 The Tagging Strategy

Traditional model labels the intent and the words slot separately as Table 1 shows. The labels of intent and word slot are divided into two collections.

Table 1. The word slots and intent of a sentence instance in ATIS corpus.

Sentence	Flights	From	Boston	To	Kansas	City	On	Friday
Word slot	O	O	B-fromloc	O	B-toloc	I-toloc	O	B-depart time
Intent	Flight							

In order to avoid the redundant labeling results and propagation interaction, we adopt a new tagging strategy. How the results are tagged is shown in Fig. 1. Based on our tagging strategy, each word is assigned to a tag that contains three parts: the word position in word slot, the word slot type, and the intent of the whole sentence. With the symbol "O" at the head of the tag, this represents the "Other" tag, which means that the corresponding word is not in any of the word slots. In addition to symbol "O", we apply the "BIES" symbol to represent the position information in word slot. The word slot type is obtained from a predefined set. The intent type symbol can also get from a predefined set but the intention of all words in a given sequence is exactly the same. Thus, the total number of tags is $N_t = N_p * N_s * N_i - \Phi$, where N_p is the number of the "BIES" position information symbol, N_s is the size of the word slot set, N_i is the number of all intents and Φ is the number of redundant labels.

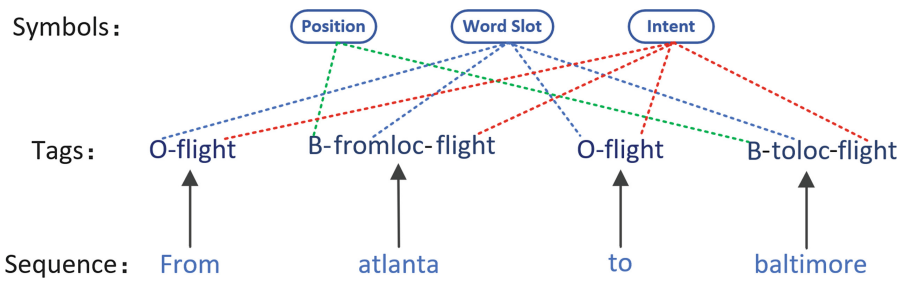

Fig. 1. The instance of our tagging strategy. The word slot symbol "fromloc" and "toloc" represent the departure and destination of the flight. the "flight" symbol expresses the intent of asking for flight information.

As is shown in Fig. 1, the word "atlanta" is signed the tag "B-fromloc-flight". The position information is marked as "B", the word slot type is marked as "fromloc", the intent type is marked as "flight" and the three parts of the tag are connected by the symbol "-". The intent of a sentence is obtained from the majority intent symbols of all the words.

3.2 Attention-Based RNN Model

In recent years, end-to-end model based on recurrent neural network has been widely used in sequence labeling task [12, 20]. In this paper, we investigate an end-to-end model to produce the extraction results as Fig. 2 shows. It contains an embedding layer, a bi-directional RNN layer and a hidden layer with attention mechanism.

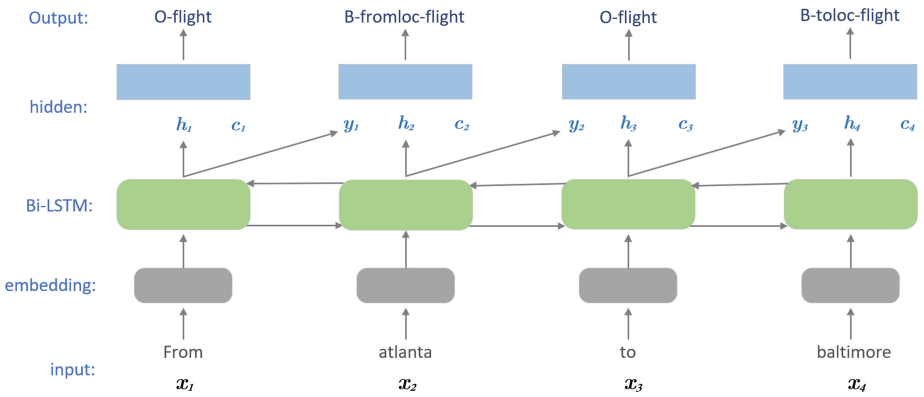

Fig. 2. The illustration of our model with a word embedding layer, a bi-LSTM layer and a hidden layer. y_i is the hidden layer output, h_i is the hidden layer state and c_i is the attention context vector.

The Bi-RNN Layer. In the sequence labeling task, we generally learn a function $f : X \rightarrow Y$ that maps the input sequence to its corresponding label sequence explicitly aligned to the given the input sequence $X(x_1, x_2, \cdots, x_T)$ and its corresponding label sequence $Y(y_1, y_2, \cdots, y_T)$. In our joint task, we want to find the best label sequence Y given input words X such that:

$$\hat{y} = \arg max\, P(Y|X) \tag{1}$$

The bidirectional RNN model has been proven to capture the semantic and sequential information for each word effectively in sequence tagging task by reading sentences bidirectionally. In our proposed model, we use a bidirectional RNN layer reading the input sequence in both forward and backward directions. The forward RNN reading the input sequence in its original order generates a hidden state fh_i at each time step i. Similarly, the backward RNN reading the input sequence in its reverse order generates a sequence of hidden states $[bh_1, bh_2, \cdots, bh_T]$. The bidirectional RNN layer hidden state h_i at each time step i is combined of the forward state fh_i and backward state bh_i, $h_i = [fh_i, bh_i]$. Each hidden state h_i carries information of the entire input sequence with strong focus on the parts around the i th word. The hidden state h and the bi-RNN output y are then combined with an attention context vector c to produce the label distribution.

The Attention Mechanism. Attention mechanism can be regarded as the process of selectively filtering a small amount of important information from all the provided information ignoring most of the non-important information [18]. The process can be reflected in the calculation of the weight coefficient. The greater the weight is, the more it focuses on its corresponding value. The weight represents the importance and the value is its corresponding information. In the joint extraction task, the attention mechanism can provide the classifier with global attention information by giving different weights to the words.

The attention mechanism is applied in a hidden layer above the bi-RNN layer. We initialize the hidden layer state using the last hidden state of the bi-RNN layer following the approach in [2]. At each time step i, the hidden layer state s_i is calculated as a function of the previous bi-RNN output y_{i-1}, the bi-RNN hidden state h_i and the attention context vector c_i:

$$s_i = f(y_{i-1}, h_i, c_i) \tag{2}$$

The attention context distribution c is generated by the hidden state h of the bidirectional RNN. In detail, c_i is calculated as the weighted sum of the bi-RNN states $h = (h_1, h_2, \cdots, h_T)$ [2]:

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \tag{3}$$

$$\alpha_{i,j} h_j = \frac{exp(e_{i,j})}{\sum_{k=1}^{T} exp(e_{i,k})} \tag{4}$$

$$e_{i,k} = g(s_{i-1}, h_k) \tag{5}$$

where g is a feed-forward neural network. The attention context vector c_i provides additional information to the hidden layer that can be viewed as weighted sequential features of the RNN hidden layer states $(h_1, h_2, \cdots, h_T)$. In this way, the attention mechanism can provide global weighted information to generate labels.

The Bias Loss Function. In order to enhance the influence of word slots we tried to use the RMSprop optimization method [15] by defining the loss function as:

$$L = max \sum_{j=1}^{|D|} \sum_{i=1}^{T} (1 + \alpha I(O)) log(p_i = y_i | x_i, \theta) \tag{6}$$

$$p_i^{(j)} = \frac{exp\left(o_i^{(j)}\right)}{\sum_{k=1}^{N_t} exp\left(o_i^{(k)}\right)} \tag{7}$$

Where $|D|$ is the size of the data set, T is the length of the sequence, y_i is the label of the i th word, p_i is the normalized probability of the tags which is defined in formula 7. N_t is the total number of tags, o_i is the output of the i th word, α is the bias weight of the

loss function. The larger α is, the more influence the corresponding tag has. $I(O)$ is a binary function that distinguishes the loss of tag "O" and word slot tags and it was defined as follows:

$$I(O) = \begin{cases} 0, & \text{tag} = \text{O} \\ 1, & \text{tag} \neq \text{O} \end{cases}$$

4 Experiments

For a better comparison with previous methods and presenting the effect of our method, we carried out experiments on the Air Travel Information System (ATIS) pilot corpus. Then our model was compared with the previous pipeline and joint training models to demonstrate the performance in both independent and joint tasks.

4.1 Experimental Settings

Dataset. ATIS (Airline Travel Information Systems) data set [4] is widely used in intent detection and word slot extraction task. The data set contains the conversation text of persons who made the flight reservation. In this work, we follow the ATIS corpus setup used in [9, 11, 16, 19]. There are 4978 conversation text from the ATIS-2 and ATIS-3 corpora in the training set and 893 conversation text from the ATIS-3 NOV93 and DEC94 data sets in the test set. The total number of word slot labels is 127 and the size of intent types is 18. We use the F1 score to evaluate the results on word slot extraction and evaluate the performance of intent detection by using classification accuracy rate.

Hyperparameters. In our experiments, LSTM cell is used as the basic RNN unit. Our LSTM implementation follows the design in [21]. The number of cells in the LSTM layer is 128. We set the initial LSTM forget gate bias as 1 [6]. In our model, there is only one LSTM layer and the multilayer LSTM will be explored in future work. The word embeddings dimension is set to 128. We randomly initialize the word embeddings and fine-tuned during backward propagation. The training batch size is 16. Dropout rate on the fully connected network is set to 0.5 for regularization [21]. To prevent the gradient from exploding, the maximum value of gradient clipping is set to 5. The bias of the loss function is set to 10 and the number of headers of the attention is set to 10. We apply Adam optimization to our model following the settings in [8].

4.2 Intent Detection Task Results

We first report the results on independent tasks of intent detection and word slot extraction. We used the bi-LSTM model as our baseline and compared the performance of our proposed model with previously reported methods on intent detection task and illustrate the results in Table 2.

As we can see, our joint methods performs better than pipelined methods on intent detection. The attention-based bi-LSTM joint model and the bi-LSTM joint model with

Table 2. The results on independent task of intent detection

Model	Intent accuracy (%)
Recursive NN [2]	95.40
Boosting [16]	95.62
Boosting + Simplified sentences [17]	96.98
bi-LSTM	97.14
bi-LSTM with attention	97.31
bi-LSTM with bias loss function	97.20
bi-LSTM with attention-bias loss function	**97.65**

bias loss function advances the bi-LSTM model. Moreover, the bi-LSTM model combined with attention mechanism and bias loss function achieved the best accuracy of intent detection. This could be attributed to the combination of attention mechanism and bias loss function that allows the model to learn the sequence level information more efficiently.

While training the attenuation model, we found the attention mechanism is helpful to enhance the influence of long-distance keywords when the intent of words is been labeling. As shown in Fig. 3, We can find that the attention weights at the beginning of the sentence are higher when we label the last word "thursday". The word slot of "thursday" is a date slot which may appear in many sentences with different intents. So we should know the intent of the sentence as well as the slot of the word "thursday" and then label the word with "B-depart_date-flight". Obviously, the beginning words carry most information of the intent and the attention mechanism can find additional long-distance information effectively to solve multiple intent issues. This may explain one side of the reason for the good performance of our joint model on intent detect task.

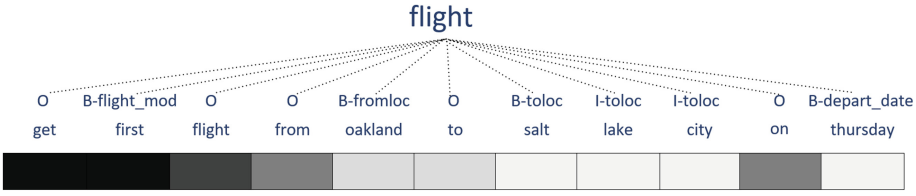

Fig. 3. The distribution of the attention weights when labeling the last word "thursday" with the intent "flight" of the sentence. The darker shade is the higher attention weight is.

4.3 Word Slot Extraction Task Results

Table 3 shows the performance of our proposed model for word slot extraction and previously reported results. Once again, the joint model performs better than the pipeline method. Besides, the attention-based model gives slightly better F1 score than the non-attention-based models. The reason could be the attention mechanism seeking to find other supporting information from input word sequence for the word slot label

prediction. Overall, attention-based RNN Models outperform the ones without attention mechanism and the bias loss function is helpful for the word slot extraction.

When we combine attention mechanism and bias loss function on bi-LSTM model we find the F1 score gets slight reduction. We think the weight of bias loss function may disrupt the weight of attention during backpropagation. As the bias is manually set, it is difficult to select a perfectly suitable hyperparameter, which may lead to human errors affecting the training process. In the next work, we will try to optimize the bias by setting it as a parameter of the model.

Table 3. The results on independent task of word slot extraction

Model	F1 score (%)
CNN-CRF [19]	94.35
RNN with Label Sampling [9]	94.89
Hybrid RNN [11]	95.06
Deep LSTM [20]	95.08
bi-LSTM	94.89
bi-LSTM with attention	**95.15**
bi-LSTM with bias loss function	95.13
bi-LSTM with attention-bias loss function	95.05

4.4 Joint Task Results

Table 4 shows our tagging model's performance on joint extraction task of intent and word slots comparing to previously reported results.

Table 4. The results of joint task on intent detection and word slot extraction

Model	F1 score (%)	Intent accuracy(%)
RecNN [2]	93.22	95.40
RecNN + Viterbi [2]	93.96	95.40
bi-LSTM	94.89	97.09
bi-LSTM with attention	**95.15**	**97.20**
bi-LSTM with bias loss function	95.13	96.89
bi-LSTM with attention-bias loss function	95.05	97.09

As shown in this table, the joint model using tagging strategy achieved promising performance on both intent detection and word slot extraction. The attention based bi-LSTM get the best performance during our experiments. However, the combination model based on attention mechanism and bias loss function still have much room for improvement.

We checked the badcase in the results, most of which were caused by the word "UNK" which represents low frequency words. Besides, many word slots are also infrequent in the mislabeling results. It can be speculated that due to the limit of the data size, the training data could not cover all the cases well, especially for words and word slots with low frequency. In future missions, we will scale the size of the data set and adopt a deeper RNN model to further improve the performance of our model.

The experimental results show the effectiveness of our proposed method. But it still has shortcoming on identifying multiple tags. In the next work, we will replace the softmax function in the output layer with multiple classifier, so that a word can be labeled multiple tags. In this way, the word tagging process can be transformed into a multi-classification problem, which can solve the problem of multiple tags. Although, our model can enhance the effect of word slot words, the associations between word slots and sentence intent still require refinement in next works.

5 Conclusion

In this paper, we explored a tagging strategy and investigated the end-to-end RNN models to jointly extract of intent and word slots. We further improved our joint tagging strategy model with the attention mechanism to solve the problem of diversified relationship between word slots and intentions. Based on our tagging strategy model, the joint task of intent detection and word slot extraction is greatly simplified as only one sequence tagging model needs to be trained and deployed. We conduct experiments on a public dataset and the experimental results show that our joint model achieved better performance on the benchmark ATIS task compared with most of the existing pipelined and joint models for both independent and joint extraction task.

Acknowledgement. This work was supported by the National Key Research and Development program of China (No. 2018YFB1004703).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Guo, D., Tur, G., Yih, W., Zweig, G.: Joint semantic utterance classification and slot filling with recursive neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 554–559. IEEE (2014)
3. Haffner, P., Tur, G., Wright, J.H.: Optimizing SVMs for complex call classification. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP 2003), vol. 1, pp. I–I. IEEE (2003)
4. 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, 24–27 June 1990 (1990)
5. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)

6. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning, pp. 2342–2350 (2015)
7. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Liu, B., Lane, I.: Recurrent neural network structured output prediction for spoken language understanding. In: Proceedings of the NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions (2015)
10. McCallum, A., Freitag, D., Pereira, F.C.: Maximum entropy markov models for information extraction and segmentation. In: ICML, vol. 17, pp. 591–598 (2000)
11. Mesnil, G., et al.: Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(3), 530–539 (2015)
12. Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531. IEEE (2011)
13. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding. In: Eighth Annual Conference of the International Speech Communication Association (2007)
14. Sarikaya, R., Hinton, G.E., Ramabhadran, B.: Deep belief nets for natural language call-routing. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5680–5683. IEEE (2011)
15. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
16. Tur, G., Hakkani-Tür, D., Heck, L.: What is left to be understood in ATIS? In: 2010 IEEE Spoken Language Technology Workshop (SLT), pp. 19–24. IEEE (2010)
17. Tur, G., Hakkani-Tür, D., Heck, L., Parthasarathy, S.: Sentence simplification for spoken language understanding. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5628–5631. IEEE (2011)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
19. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 78–83. IEEE (2013)
20. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 189–194. IEEE (2014)
21. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
22. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075 (2017)