Attention-Based CNN-BLSTM Networks for Joint Intent Detection and Slot Filling

Yufan Wang, Li Tang, and Tingting He(✉)

School of Computer, Central China Normal University, Wuhan 430079, China
{yufan_wang,lt}@mails.ccnu.edu.cn,
tthe@mail.ccnu.edu.cn

Abstract. Dialogue intent detection and semantic slot filling are two critical tasks in nature language understanding (NLU) for task-oriented dialog systems. In this paper, we present an attention-based encoder-decoder neural network model for joint intent detection and slot filling, which encodes sentence representation with a hybrid Convolutional Neural Networks and Bidirectional Long Short-Term Memory Networks (CNN-BLSTM), and decodes it with an attention-based recurrent neural network with aligned inputs. In the encoding process, our model firstly extracts higher-level phrase representations and local features from each utterance using convolutional neural network, and then propagates historical contextual semantic information with a bidirectional long short-term memory network layer architecture. Accordingly, we could obtain sentence representation by merging the two architectures mentioned above. In the decoding process, we introduce attention mechanism in long short-term memory networks that can provide additional sematic information. We conduct experiment on dialogue intent detection and slot filling tasks with standard data set Airline Travel Information System (ATIS). Experimental results manifest that our proposed model can achieve better overall performance.

Keywords: Nature language understanding · Slot filling · Intent detection
Attention model

1 Introduction

Intelligent dialogue system is a core technology in artificial intelligence and will become a friendly human-computer interaction approach. Building dialogue systems usually requires great effort, because the training of NLU and speech generation modules are essential to conduct conversations [1]. NLU, as a critical component of the dialogue system, is a major problem in current intelligent voice interaction and human-machine dialogue. The purpose of NLU is to convert user dialogue text into a form that computer can understand. Typically, NLU system involves the identification of user intent from natural language queries and the extraction of semantic constituents. Generally, the two tasks are referred as intent detection and slot filling [2].

Usually, intent detection and slot filling are performed separately. Intent detection can be abstracted as a classification problem. Slot filling can be abstracted as a sequence labeling problem. There are some traditional methods based on statistics used

M. Sun et al. (Eds.): CCL 2018/NLP-NABD 2018, LNAI 11221, pp. 250–261, 2018.
https://doi.org/10.1007/978-3-030-01716-3_21

for both tasks. However, the traditional methods are not capable of learning the deep semantic information. Deep neural networks have brought new inspiration to various natural language processing tasks, including intent detection and slot filling. In intent detection task, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two popular models. While RNN and encoder-decoder have achieved good results in slot filling [2, 3]. Nowadays, there are many joint models used to solve intent detect and slot filling tasks in the meantime. And the joint models achieved good performance in two tasks.

Encoder-decoder neural network is used to solve the slot filling and performs very well in [2]. Generally, sentence representation obtained from encoder based RNN only considers the chronological features of sentences but ignores the importance of local semantic information of sentences. In task of slot filling, however, local semantic information has important influence on identifying the slot correctly. Fortunately, convolution layer is specialized in capturing local semantic information [4], and it can get higher-level representation of sentences. Therefore, we explore CNN results input into the RNN for encoding and get the final sentence representation. In order to keep the original chronological order in the RNN input, we rearrange the output of convolution layer in the relative order of the original sentence. Intuitively, each word between the input and output in the slot filling task is a one-to-one correspondence. Attention mechanism and alignment could obtain corresponding relationship. The sentence representation obtained by encoder can also be used for intention detection. The model joint the two tasks can make them influence and promote each other [5]. In this paper, a new joint model is presented to solve intent detection and slot filling tasks. In the new joint model, a combination of CNN and bidirectional long short-term memory networks (BLSTM) encodes a dialogue sentence into a dense vector that imply features of the sentence. Then, this vector is used to initialize the decoder state. The decoder uses long short-term memory networks (LSTM) cell as basic RNN. User intent and slot labels can be generated by decoding. In addition, an attention-based RNN with aligned inputs is used to decode. Experimental results manifest that our proposed joint model can achieve better overall performance.

The remainder of the paper is organized as follows. Section 2 discusses related work. The details of our model and the proposed methods are described in Sect. 3. Section 4 describes experimental setup. In Sect. 5, the experiment results and analysis are reported. Finally, Sect. 6 draws conclusions and discusses future work.

2 Related Work

In natural language processing, NLU has been a hot research field. NLU typically involves two major tasks—intent detection and slot filling. Intent detection can be abstracted as a classification problem. There are some popular classifiers based on supervised machine-learning methods like Support Vector Machine (SVM) [6], Nave Bayes [7] and so on. Deep learning methods recently have been shown to give excellent performance in natural language processing field. Grabes et al. [8] compared feedforward, recurrent, and gated—such as LSTM and GRU (Gated Recurrent Unit)—networks for intent classification task. Xiao et al. [9] proposed a neural network

architecture that connect both the convolution and recurrent layers for intent detection. Slot filling can be abstracted as a sequence annotation problem. The popular traditional approaches include Maximum Entropy Markov Model (MEMM) [10], Conditional Random Field (CRF) [11]. Recently, Aliannejadi et al. [12] proposed graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data. Although traditional machine learning methods have achieved good results, these methods require good feature engineering and additional semantic resources. In recent years, RNN, CNN, as well as their variations or combinations are used to solve slot filling problems. Xu et al. [13] proposed using CNN based triangular CRF for joint intent detection and slot filling. Yao et al. [14] proposed a regression model on top of the LSTM unnormalized scores for slot filling. Vu et al. [15] proposed bi-directional recurrent neural network with ranking loss for slot filling. Kurata et al. [16] proposed leveraging sentence-level information with encoder LSTM for slot filling. Zhu et al. [17] proposed encoder-decoder with focus-mechanism for slot filling. And the combination of LSTM and CRF achieved good results in slot filling.

Joint model for intent detection and slot filling has also been proposed last several years. Guo et al. [18] proposed using a recursive neural network that learns hierarchical representations of the dialogue for the joint task. Specially, an encoding-decoding joint model is used to solve intent detection and slot filling [19]. Liu [2] proposed an attention-based encoder-decoder neural network models for intent detection and slot filling. Zhang et al. [5] proposed using Bidirectional Gated Recurrent Unit (BGRU) for the joint task. Weigelt et al. [20] proposed jointly modeling intent identification and slot filling with contextual and hierarchical information. Such joint models simplify intent detection and slot filling systems, as only one model needs to be trained and deployed. In our work, we have made some changes on the encoder-decoder architecture and integrated CNN with BLSTM as the encoder. Combining the strengths of CNN and LSTM in encoder, we propose a joint model for intent detection and slot filling.

3 Methodology

Our model is inspired by the works of Liu et al. [2], Zhou et al. [21], and Yao et al. [22]. The architecture of the Encode-decoder model is shown in Fig. 1. The main structure includes: encoder (input layer, convolutional layer, window feature sequence and a bidirectional recurrent structure of LSTM) and decoder (LSTM network with attention mechanism and aligned inputs). In terms of encoder, the structure which combines CNN and BLSTM was proved to be more powerful than single RNN in our experiment. In particular, we choose the BLSTM network. Compared to vanilla RNN, BLSTM has the ability to better model long-term dependencies. As for decoder, on one channel, the result of intent detection can be obtained through a softmax function; on the other channel, a LSTM network with attention mechanism and aligned inputs can be used for slot filling.

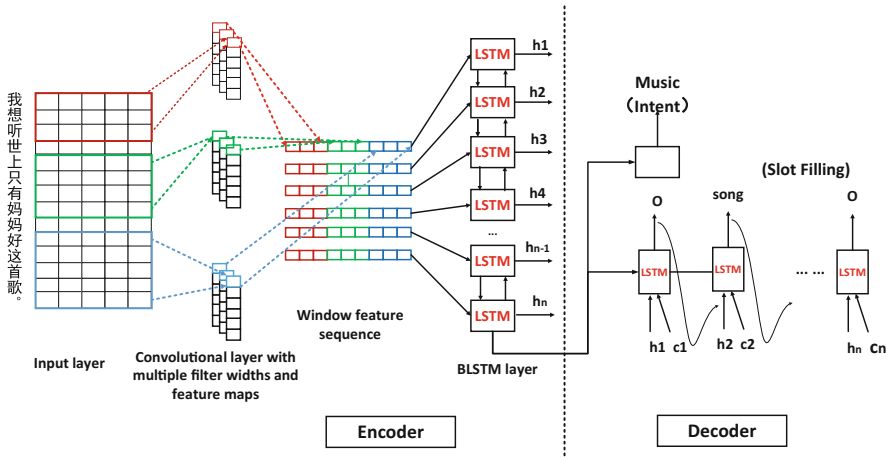

Fig. 1. The architecture of the encoder-decoder model is shown above. The word embedding is used as input to the model. Rearrange the feature maps obtained from convolution layer then feed them to BLSTM. The last state of the encoder is used to initialize the decoder state. An attention-based RNN with aligned inputs is used to decode.

3.1 Encoder

The normal encoder uses LSTM as the encoding network. The improved encoder structure is shown in Fig. 1. In this section, we will introduce the functions and calculation methods for each layer of encoder.

Embedding Layer. In the first step, the dialogue text needs to be converted into a feature vector matrix as the input layer of the model. This model uses embedding technology to convert each word feature into a word embedding vector. We can get the word embedding vectors which we need by using word embedding tools (e.g. Word2vec [23] by google). So we get dialogue text vectors of input $X = [x_1, x_2, x_3, \ldots, x_n]$.

Convolutional Layer. The CNN model has been proved to perform well on some tasks in the NLP field. Convolutional layer of CNN can encode significant information contained in input data with significantly fewer parameters [24]. Convolutional layer is equivalent to a sliding window, which can get the contextual characteristics within the local window of the current word. The parameters of each filter are shared across all the windows. In intent detection and slot filling tasks, we find that a good or comparable performance can be achieved with only one convolutional layer. Multiple filters with differently initialized weights are used to improve the model's learning capability. Generally, multiple filter sizes perform better than a single size. The number of filters k is determined to use experience value. The convolution operation is as follows:

$$s_i = h\left(W_S^T \cdot x_{i:i+m-1} + b_s\right) \tag{1}$$

where Ws $\in \mathbb{R}^{m\times \mathrm{d}}$, d is the word embeddings' dimension, and m is the size of convolution window. The vector $x_{i:i+m-1}(x_i \in X = [x_1, x_2, x_3 \ldots x_n])$ represents a window of m words starting from the i-th word to $(i+m-1)$-th word, and the $x_i \in \mathbb{R}^d$, the term b_s is a bias vector. $h(\cdot)$ denotes a nonlinear activation function. The outputs of the feature maps is: $S = [s_1, s_2, s_3 \ldots s_n]$, where n is the number of convolution windows, s_i is the result of every convolution.

Window Feature Sequence. To learn sequential correlations from higher-level sequence representations, we do not feed the output of the convolutional layer directly into the BLSTM. Instead, we add a structure, namely the window feature sequence layer, before feeding the output into the BLSTM. We get the outputs $(S = [s_1, s_2, s_3 \ldots s_n])$ of the convolution layer and rearrange it in the relative order of the original sentence to get the vector $V = [v_1, v_2, v_3 \ldots v_l]$. The window feature sequence is connected by the elements corresponding to the i-th dimension of each feature map after convolution.

$$v_i = f_1^i \oplus f_2^i \oplus f_3^i \oplus \cdots f_{n-1}^i \oplus f_n^i \tag{2}$$

Here, f_n^i denotes the elements corresponding to the i-th dimension of the s_n. The length of feature map and the input length of convolution layer are equal to l. So a total of l combinatorial vectors can be obtained. Each sentence representative is transformed into successive window features. Typically, the pooling operation would be performed after the convolutional layer, but the discontinuous feature selection sampling will destroy the sequential information of the sentence. While, the window feature sequence layer can effectively maintain the sequential information, which contributes to the effective extraction of various features of a given sentence.

Sentence Representation with BLSTM. RNN is able to propagate historical information via a chain-like neural network architecture. Therefore, we choose to use RNN network after the convolution layer instead of max-pooling. In RNN, gradient vanishing and gradient exploding are two problems which are difficult to solve when propagating backwards in the time series [25]. Graves A et al. [26] designed LSTM in order to overcome the problem. An adaptive gating mechanism is introduced to help LSTM unit to keep the previous state and remember the extracted features of the current input data.

A basic LSTM neural network consists of four modules: input gate i_t, forget gate f_t, output gate o_t and memory cell c_t. The following formulas show how to calculate each gate and memory cell unit:

$$i_t = \sigma(W_i \cdot v_t + U_i \cdot h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma\left(W_f \cdot v_t + U_f \cdot h_{t-1} + b_f\right) \tag{4}$$

$$o_t = \sigma(W_o \cdot v_t + U_o \cdot h_{t-1} + b_o) \tag{5}$$

$$g_t = tanh\big(W_g \cdot v_t + U_g \cdot h_{t-1} + b_g\big) \tag{6}$$

Here $v_t (v_t \in V = [v_1, v_2, v_3 \ldots v_l])$ denotes the current input, and h_{t-1} denotes the previous hidden state. W, U are weight matrices for input and output, respectively, and b is the corresponding bias.

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1} \tag{7}$$

$$h_t = o_t \otimes tanhc_t \tag{8}$$

Where g_t is the new memory content, c_t is updated by partially forgetting the existing memory and adding a new memory content. Here we use tanh as our non-linear activation function.

According to the LSTM correlation formula, we can deduce the formula of the forward and the backward LSTM. For each timestep t in the input sequence, we will get two kinds of sentence representations $\overleftarrow{h}_t, \overrightarrow{h}_t$. We can obtain the outputs of each timestep in the BLSTM layer:

$$h_t = \left[\overleftarrow{h}_t, \overrightarrow{h}_t\right] \tag{8}$$

The encoder output of sequence: $H = [h_1, h_2, h_3 \ldots h_n]$

3.2 Decoder

The decoder for intent detection and the decoder for slot filling share the same encoder. During model training, we add the two loss functions together as the final cost function. Costs from both decoders are back-propagated to update the model parameters.

Decoder for Intent Detection. We add an additional decoder for intent detection task that shares the same encoder with slot filling decoder. For intent detection, we can get the last output o_0 of encoder, which encodes information of the entire source sequence.

$$y_i = W \cdot o_0 + b \tag{9}$$

where o_0 is also initial decoder state, we use the softmax function to convert the output data into probabilities p_i as follow:

$$p_i = \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)} \tag{10}$$

We get the category of max probabilities as the intention of dialogue text

Attention-Based Decoder for Slot Filling. In our model, attention mechanism is introduced to improve the model effect. The attention mechanism can automatically select some important information of word sequence. And attention mechanism let the task-processing system focus more on finding useful information related to the current output in the input data. The ultimate goal of the attention model is to help a framework, such as encoder-decoder, to better learn the interrelationships between multiple content modes and to better represent this information. o_i is the decoder state of the slot label predicted in time step i. The formula is the calculation process of decoder state o_i:

$$o_i = f(o_{i-1}, y_{i-1}, h_i, c_i) \tag{11}$$

where o_{i-1} is the previous decoder state, and y_{i-1} is the previous emitted label. h_i is the encoder hidden state to introduce aligned information. And the context vector c_i is computed as a weighted sum of the encoder states $H = [h_1, h_2, h_3 \ldots h_n]$ as follow:

$$\alpha_i = \frac{\exp(\eta(h_i, o_{i-1}))}{\sum_{i=1}^{n} \exp(\eta(h_i, o_{i-1}))} \tag{12}$$

and

$$\eta(h_i, q) = Wh_i + Uo_{i-1} \tag{13}$$

$$c_i = \Sigma_{i=1}^{n} \alpha_i h_i \tag{14}$$

where W and U. are parameters. Compared to the decoder without using attention mechanism, the context vector c_i. provides more information for decoder.

4 Experiments

4.1 Data Set

In order to verify the validity of our model, we conduct dialogue intent detection and slot filling on Airline Travel Information Systems (ATIS) dataset [27], which is widely used in NLU research. ATIS is split into 4978 utterances for training and 893 utterances for test. There are 127 slot labels and 22 intent types in the data set. And the ATIS data set has an average of 15 words per sentence. F1-score is used as evaluation metric for slot filling, and accuracy is used as evaluation metric for the task of intent detection.

4.2 Model Training

We explore the effects of different hyper parameters in the proposed method: word embedding size, learning rate, number of filters, batch size, window size for convolution, the hidden layer size and so on.

The hyper-parameter settings of the neural networks may have a great impact on the final results of the experiment, especially the filter sizes in CNN. We discuss the four different sizes of 2, 3, [2, 3, 5] and [2, 4, 5] about convolution kernels, and the sizes of [2, 3, 5] performs the best. The default forget gate bias is set to 1 in LSTM network [28]. Adam optimization algorithm is adopted to calculate the model. In this work, other hyper-parameters are shown in Table 1.

Table 1. Hyper-parameters of our model

Embedding size	64
Hidden units	128
Learning rate	0.001
Loss regularization	0.001
Window size	[2, 3, 5]
Filter number	128
Dropout	0.5

4.3 Compared Methods

Standard neural network model LSTM and the CRF model are chosen for the comparative experiment. The specific method for each model is designed as follows

- CRF: We choose the standard CRF as baseline. The input is the n-grams in a context window. The baseline was demonstrated in the ATIS dataset.
- CNN-CRF: The architecture combination of CNN and CRF, which exploits the dependency between intents and slots. The two tasks to share the same features extracted through CNN layers.
- Deep LSTM: We select a deep LSTM network which consists of multiple layers of LSTMs for comparison.
- RecNN: The model adapted recursive neural networks for joint training. To improve the result on slot filling, the Viterbi algorithm was introduced.
- Bi-RNN with attention: We select bidirectional recurrent neural network with attention mechanism for comparison.
- Bi-RNN with Ranking Loss: We select bidirectional recurrent neural network for comparison. Furthermore, the model use ranking loss function to train the model.
- Encoder-laber Deep LSTM: encodes the whole input sequence into a fixed length vector and then uses this encoded vector as the initial state of another LSTM for sequence labeling.
- Encoder + Decoder: We choose BLSTM as encoder and a unidirectional LSTM as decoder. Attention mechanism and aligned inputs are introduced in decoder.
- Encoder + Decoder (Focus): We choose a standard bidirectional LSTM as encoder and a unidirectional LSTM as decoder. Focus mechanism which is for better alignment is introduced in decoder.

5 Experimental Results and Analysis

Table 2 shows the results of our joint model and previous reported models. Compared with other standard neural networks on the same dataset, our model outperforms both on precision of intent detection and F1-score of slot filling. We discuss the advantage of using BLSTM-CNN as an encoder to encode the input sequence for intent detection and slot filling in task-oriented dialog systems in the next experimental analysis.

Table 2. Comparison with published results on ATIS.

Model	ID precision	SF F1
CRF	–	92.94%
Simple RNN	–	94.11%
CNN-CRF [13]	–	94.35%
LSTM	–	94.85%
Deep LSTM [14]	–	95.08%
Bi-RNN with ranking loss [15]	–	95.56%
Encoder-labeler Deep LSTM [16]	–	95.66%
BLSTM-LSTM (focus) [17]	–	95.79%
RecNN [18]	95.40%	93.22%
Bi-GRU & max-pooling [5]	96.98%	95.61%
BRNN (Attention) [2]	96.65%	95.78%
Encoder-decoder [2]	96.57%	95.87%
Our model	**97.17%**	**97.76%**

5.1 Our Model vs. Other Joint Model

In Table 3, compared with the previous joint model, the F1-score of slot filling and precision of intent detection are both higher in our model

Table 3. Joint model result on AITS

Model	ID precision	SF F1
Encoder-decoder	96.57%	95.87%
Our model (only attention)	96.87%	95.77%
Our model (only aligned)	96.99%	96.89%
Our model (attention and aligned)	**97.17%**	**97.76%**

In the slot filling task, the results of the various models we discussed show that: First, slot filling models with no alignment information do not perform well. The reason is that the alignment between the words and labels is explicit. But the model may not be able to learn the alignment from the training data. Second, we can see that the F1 score and accuracy of the attention-based models are higher than models with no attention

mechanism. By investigating the attentional mechanism learned by the model, we found that there are differences in the distribution of attentional weights among words in the source sequence. The reason is that attention mechanism can better consider the importance of contextual information. Third, the attention-based encoder-decoder model with aligned inputs has a lower F1 score than the attention-based encoder-decoder model based on CNN-BLSTM. Because combining the bidirectional LSTM with the convolutional structure enables the model to extract comprehensive information, namely historical, future and local context of any position in a sequence. In the intent detection task, the accuracy of our model also exceeds the standard Bidirectional Recurrent neural networks. The attention-based encoder-decoder model performs better in that the model extracts the deep local semantic information of the dialogue text and the salient features of implicit category information through convolution layer. We infer that our model could find the balance between two loss functions of two subtasks. This is an important reason for our model to achieve better overall performance.

5.2 Separate Model vs. Joint Model

Table 4 shows the experiment results of joint model and separate model. First, we give the definition of joint model and separate model. The joint model we propose can simultaneously get the both results of intent detection and slot filling. The separate model is only for one task. For intent detection, the separate model is the same as the joint model in encoder, but only intent detection task is concerned in decoder. Slot filling is also the same. The joint model outperforms the single model on both tasks, indicating that joint training is effective. We add the loss functions of the two tasks as the loss function of the joint model. Through a unified loss function and a common representation, the model learns the correlation between the two tasks to promote each other. Because of the information sharing of two channels, our joint model outperforms the pipeline method with single channel. The joint model could make a balance between the two loss functions of two subtasks. In addition, the model shares the model parameters to speed up the training procedure.

Table 4. The result of model Joint and separate on AITS

Model	ID precision	SF precision	SF F1
Intent detection only	96.51%	–	–
Slot filling only	–	97.36%	97.03%
Join model	**97.17%**	**97.94%**	**97.76%**

5.3 Filter Sizes Setting

In the convolutional layer, the local feature of the dialogue text is extracted by the filter, which is similar to the N-gram feature. Different sizes of the filter bring about different local feature maps. So, the different size of filters in the convolutional layer has an important effect on convolutional results. Generally, multiple filter sizes perform better than a single size.

In Table 5, we found that single convolutional layer with filter size [2, 3, 5] outperforms the other cases in our experiments. Comparing 3, [2, 3, 5] with 2, [2, 4, 5], we can see that 3, [2, 3, 5] performs better in slot filling and intent detection, which proves that tri-gram features play an important role in capturing local features in our task. We speculate that BLSTM could learn better semantic sentence representation from the sequences of tri-gram features.

Table 5. The result of different size of filters in the convolution layer

Filter size	ID precision	SF precision	SF F1
2	96.37%	97.20%	96.69%
3	96.48%	97.52%	96.91%
[2, 4, 5]	96.57%	97.60%	97.13%
[2, 3, 5]	**97.17%**	**97.94%**	**97.76%**

6 Conclusions and Future Work

In this paper, we discuss the feasibility of joint model which uses a combination of CNN and BLSTM as an encoder and uses an attention-based LSTM with aligned inputs as decoder for intent detection and slot filling. In our model, the semantics of the context can be captured by bidirectional LSTM neural network. And the local important latent semantic factors can be selected by convolution layer. Experiments show that our model which combines the advantages of BLSTM and CNN can achieve better performance than baseline models.

Due to the particularity of the dialogue text, there are still many problems in the task, such as new word feature misrecognition and other issues. Consequently, how to construct effective word embeddings and solve new words problem in dialogue utterance will be the future work of our study.

Acknowledgments. This research is supported by the Fundamental Research Funds for the Central Universities (CCNU18JCK05), the National Natural Science Foundation of China (61532008), the National Science Foundation of China (61572223), the National Key Research and Development Program of China (2017YFC0909502).

References

1. Shen, B., Inkpen, D.: Speech intent recognition for robots (2017)
2. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling (2016)
3. Liu, B., Line, I.: Recurrent neural network structured output prediction for spoken language understanding (2015)
4. Kim, Y.: Convolutional neural networks for sentence classification. Eprint Arxiv (2014)
5. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding, pp. 5690–5694 (2016)

6. Haffner, P., Tur, G., Wright, J.H.: Optimizing SVMs for complex call classification (2003)
7. Chen, J., Huang, H., Tian, S., et al.: Feature selection for text classification with Naive Bayes. Expert Syst. Appl. Int. J. **36**(3), 5432–5435 (2009)
8. Graves, A., Jaitly, N., Mohamed, A.R.: Hybrid speech recognition with deep bidirectional LSTM (2014)
9. Xiao, Y., Cho, K.: Efficient character-level document classification by combining convolution and recurrent layers (2016)
10. Xiao, J., Wang, X., Liu, B.: The study of a nonstationary maximum entropy Markov model and its application on the pos-tagging task. ACM Trans. Asian Lang. Inf. Process. **6**(2), 7 (2007)
11. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding (2007)
12. Aliannejadi, M., Kiaeeha, M., Khadivi, S., et al.: Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data (2017)
13. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling (2014)
14. Yao, K., Peng, B., Zhang, Y., et al.: Spoken language understanding using long short-term memory neural networks (2015)
15. Vu, N.T., Gupta, P., Adel, H., et al.: Bi-directional recurrent neural network with ranking loss for spoken language understanding (2016)
16. Kurata, G., Xiang, B., Zhou, B., et al.: Leveraging sentence-level information with encoder LSTM for natural language understanding (2016)
17. Zhu, S., Yu, K.: Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding (2017)
18. Guo, D., Tur, G., Yih, W.T., et al.: Joint semantic utterance classification and slot filling with recursive neural networks (2015)
19. Liu, B., Lane, I.: Joint online spoken language understanding and language modeling with recurrent neural networks (2016)
20. Weigelt, S., Hey, T., Landhäußer, M.: Integrating a dialog component into a framework for spoken language understanding (2018)
21. Zhou, C., Sun, C., Liu, Z., et al.: A C-LSTM neural network for text classification. Comput. Sci. **1**(4), 39–44 (2015)
22. Yao, K., Peng, B., Zhang, Y., et al.: Spoken language understanding using long short-term memory neural networks. In: IEEE – Institute of Electrical & Electronics Engineers, pp. 189–194 (2014)
23. Word2vec Homepage. http://code.google.com/archive/p/word2vec/
24. Yin, W., Schütze, H., Xiang, B., et al.: ABCNN: attention-based convolutional neural network for modeling sentence pairs (2015)
25. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. Aistats (2005)
26. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks (2013)
27. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. In: Proceedings of the Darpa Speech & Natural Language Workshop, pp. 96–101 (1990)
28. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning, pp. 2342–2350. JMLR.org (2015)