Exploring Deep Learning Architectures Coupled with CRF Based Prediction for Slot-Filling

Tulika Saha(✉), Sriparna Saha, and Pushpak Bhattacharyya

Department of Computer Science and Engineering,
Indian Institute of Technology Patna, Dealpur Daulat, India
{sahatulika15,sriparna.saha,pushpakbh}@gmail.com

Abstract. Slot-filling is one of the most crucial module of any dialogue system that focuses on extracting relevant and necessary information from the user utterances. In this paper, we propose variants of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models for the task of slot-filling which includes LSTM/GRU networks, Bi-directional LSTM/GRU (Bi-LSTM/GRU) networks, LSTM/GRU-CRF and Bi-LSTM/GRU-CRF networks. Variants of LSTM/GRU is used for discourse modeling i.e., to capture long term dependencies in the input sentences. A Conditional Random Field (CRF) layer is integrated with the above network to capture the sentence level tag information. We show the experimental results of our proposed model on the benchmark Air Travel Information System (ATIS) dataset which indicate that our model performed exceptionally well compared to the state of the art.

Keywords: Dialogue system · Natural language understanding
Slot-filling · LSTM · GRU · CRF

1 Introduction

Natural Language Understanding (NLU) forms one of the most critical module of any dialogue system. Understanding the real intention of the user and to identify the relevant information from the user query - often referred to as *slot-filling*, is fundamental for any human-computer interaction. The NLU module typically consists of the following three tasks: Dialogue Act Classification (DAC), Intent Detection (ID), and slot-filling. With considerable advancement of deep learning (DL) for sentence classification such as DAC [6,7] and ID [2,18], this paper focuses on employing DL based approach to slot filling.

Slot-filling is basically searching of user texts to extract relevant information in order to fill predefined slots in a reference knowledge base [3,17]. Slot-filling is often framed as a sequence labeling task, which maps an observation sequence x $= \{ x_1, \ldots, x_T \}$ to a sequence of labels y $= \{ y_1, \ldots, y_T \}$, i.e., to acquire the most probable slot sequence given some word sequence. An example of an user

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11301, pp. 214–225, 2018.
https://doi.org/10.1007/978-3-030-04167-0_20

utterance along with its slot labels are shown in Table 1. The most extensively used idea to solve this problem is the application of Conditional Random Fields (CRFs) [8], where given the input sequence, the probability of a label sequence is computed using an exponential model. Therefore, CRF produces distinct and globally most likely label sequence and it has been applied broadly in [13,17, 20]. Machine Translation models [10] and Maximum Entropy Markov Models (MEMMs) [16,17] are some of the other sequence labeling methods that have been studied for this task. The recent growth and success of deep learning has motivated to it being employed for solving the slot-filling task as well. Some of the most notified works include [9,12,19,22,23] where variations of Recurrent Neural Network (RNN) models have been studied extensively because of their strong potential in modeling temporal dependencies. In this paper, we propose variants of RNN such as LSTM [4], Bi-LSTM, GRU [1] and Bi-GRU to incorporate past and future input features coupled with a CRF layer to model the sentence level tag information; thus, producing state of the art results for the task.

Table 1. An example utterance with its slot

Utterance	Show	me	flights	from	atlanta	to	washington
Slot	O	O	O	O	B-fromloc.city_name	O	B-toloc.city_name

The remaining of the paper is arranged as follows: Sect. 2, presents a brief description of the related works followed by the motivation and contribution of this particular work. The proposed methodology has been discussed in Sect. 3. Section 4 examines the experimental results and its analysis. Lastly, the conclusion and the course for future work are discussed in Sect. 5.

2 Related Works

This section provides a brief description of the works done so far on slot-filling followed by the motivation behind solving this problem.

2.1 Background

Different RNN architectures, including the Jordan-type and Elman-type recurrent networks and their variants were implemented in [12] on the ATIS dataset. They reported a F1-score of 93.98. In another such work, [22] implemented a variation of RNN incorporating context words as features along with some lexical and non-lexical features. They reported a F1-score of 96.60 on the ATIS dataset. In one of the works, [19] proposed a sequential convolution neural network model with previous context words as features and gives attention to current words with its surrounding context. They reported a F1-score of 95.61 on the ATIS dataset. Variants of RNN architecture were presented in [11] that

uses the objective function of a CRF, and thereby the RNN parameters are trained based on this objective function, i.e., the whole set of model parameters, including RNN parameters and transition probabilities, are trained jointly. They reported a F1-score of 96.46.

2.2 Motivation and Contribution

Identification of the correct slots can assist an automated system to produce an appropriate response thereby helping the system in resolving the queries of the user. The problem becomes more challenging and difficult when the system needs to handle more realistic, natural utterances expressed in natural language, by a number of speakers. Irrespective of the approach being adopted, the problem is the "naturalness" of the spoken language input. Though RNNs and its variants have been used extensively for slot-filling task but they didn't model label sequence dependencies explicitly. The tokens in a sentence share a dependency with each other in order to capture context information which is addressed using RNN and its variants. Based on this dependency, tags are assigned to each tokens to model this problem. Similarly, the tags assigned to each tokens share dependency with each other which can add valuable information for modeling this sequence labeling problem. Therefore, in this work, we study and assess the effectiveness of using variants of LSTM and GRU networks for slot-filling, with significant attention on modeling label sequence dependencies.

The major contributions of this work are:

- A novel LSTM/GRU network is proposed that takes in past input features coupled with the CRF layer to incorporate the sentence level tag information in order to model label sequence dependencies.
- The proposed model is extended to a Bi-directional LSTM/GRU which incorporates the information from past and future words for prediction along with the CRF layer.
- Experimental analysis of all the models have been presented in detail.

3 Proposed Methodology

In this section the proposed methodology which includes the baseline and proposed models are described in detail.

3.1 Baseline Models

Previously, approaches such as CRF, LSTM and Bi-LSTM have been used to model the task of slot-filling. With the introduction of GRU, it has also found significant attention because of its comparable performance to LSTM. Therefore, we implement each of them as our baseline models to observe their performance and influence.

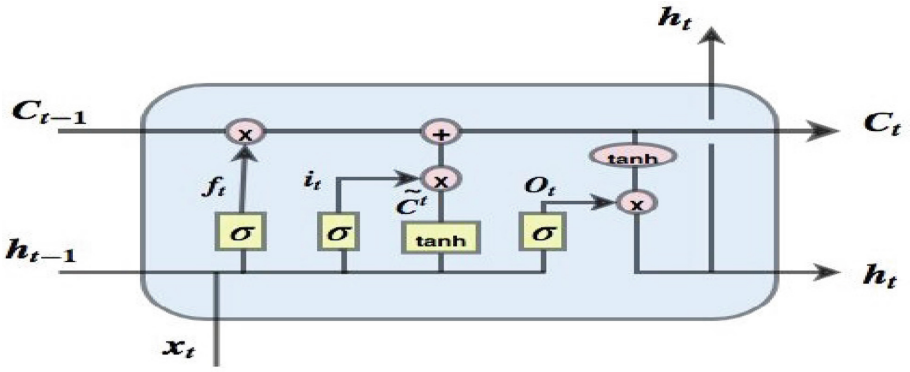

Fig. 1. A LSTM cell

– **Model 1: LSTM Networks.** LSTMs are similar to RNNs with the exception that the updates of the hidden layer in RNNs are changed by purpose-built memory cells in LSTMs. Because of which they are comparatively good in identifying and modeling long range dependencies in input data. A typical LSTM cell is shown in Fig. 1[1]. The working of the LSTM cell is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$
$$h_t = o_t * tanh(C_t) \tag{6}$$

where f_t, i_t, o_t are the forget, input and output gate, respectively. C_{t-1}, C_t are cell states at time-step $t-1$ and t, respectively. h_{t-1}, h_t represent hidden state vectors at time-step $t-1$ and t, respectively. W_f, W_i, W_o represent hidden-forget gate, hidden-input gate, hidden-output gate matrix, respectively. Logistic sigmoid function is represented by σ. Figure 2 shows a LSTM based slot-filling model which implements the above mentioned LSTM cell at its core. Pre-trained word embeddings have been used to represent input words as word vectors. The output represents a probability distribution over labels at time t.

– **Model 2: GRU Networks.** GRUs are similar to LSTMs but the key difference is that a LSTM has three gates particularly forget, input and output gates whereas GRU has two gates which are reset and update gates. Analogous to the LSTM unit, the GRU unit also supervises the flow of information, but does so without using a memory unit. It simply unmasks the entire hidden content without any restriction. The performance of GRU is comparable

[1] https://isaacchanghau.github.io/post/lstm-gru-formula/.

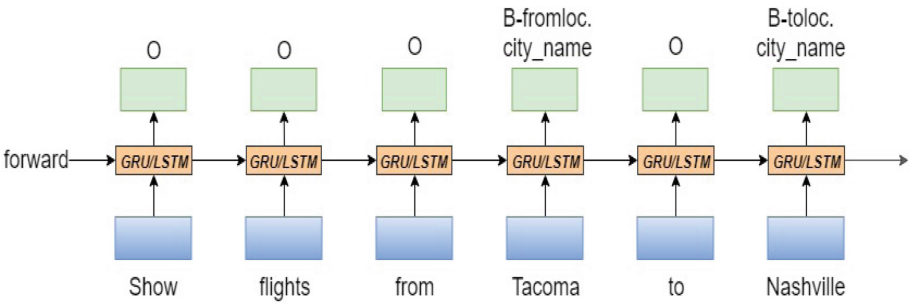

Fig. 2. A LSTM/GRU network

to that of LSTM, but it is computationally more efficient. Figure 3[2] shows a typical GRU cell. The working of the GRU unit is as follows:

$$Z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{8}$$

$$\tilde{h_t} = tanh(W \cdot [r * h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{10}$$

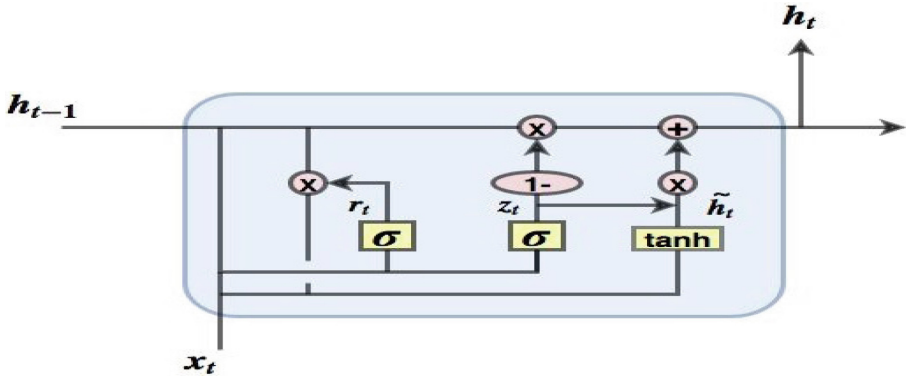

Fig. 3. A GRU cell

where z_t and r_t are update and reset gates, respectively. h_{t-1}, h_t represent hidden state vector at time-step $t-1$ and t, respectively. W_z, W_r represent hidden-update gate, hidden-reset gate matrix, respectively. Logistic sigmoid function is represented by σ . Figure 2 shows a GRU based slot-filling model which implements the above mentioned GRU cell at its core.

[2] https://isaacchanghau.github.io/post/lstm-gru-formula/.

– **Model 3: Bi-directional LSTM/GRU Networks.** Use of LSTM/GRU units provides access to just past input features. Thus, utilizing a bi-directional LSTM/GRU networks provides access to both past (through forward states) and future (through backward states) input features for a particular time frame. Figure 4 shows a bi-directional LSTM/GRU based slot-filling model.

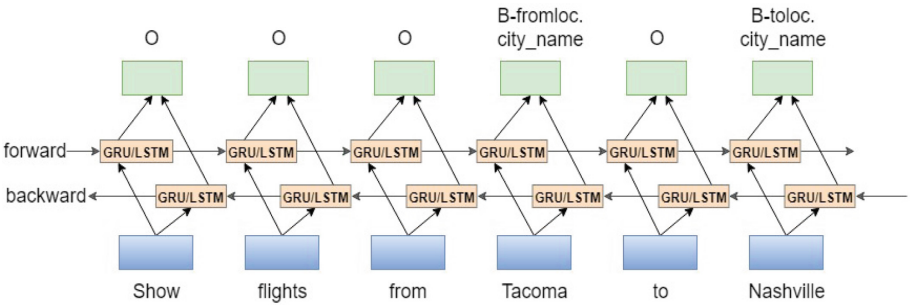

Fig. 4. A bi-directional LSTM/GRU network

– **Model 4: CRF Networks.** A basic CRF model has been implemented with input word and its Part-of-Speech tag[3] as features. Figure 5 shows a CRF based slot-filling model. CRFs work on sentence level rather than individual position; thus, taking the context into account. CRFs, in general have been seen to perform reasonably good for sequence labeling task.

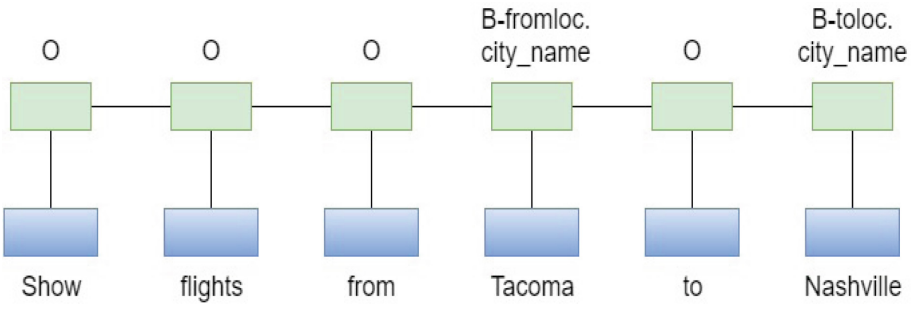

Fig. 5. A CRF network

[3] Used Stanford PoS tagger https://nlp.stanford.edu/software/tagger.shtml.

3.2 Proposed Models

– **Model 1: LSTM/GRU - CRF Networks.** This particular approach combines a LSTM/GRU network with a CRF network to obtain a LSTM/GRU-CRF model as shown in Fig. 6. The idea behind such an approach is that this network can then efficiently utilize the past input features because of the presence of the LSTM layer followed by a CRF layer which can then add sentence level tag information. CRF layer is shown by lines that joins successive output layers thus, predicting the current tag with the help of past and future tags which is quite similar to a bi-directional LSTM/GRU network that makes use of past and future input features. The output from the network is considered to be a matrix of scores say $f_\theta([y]_1^K)$. Therefore, the item $[f_\theta]_{(i,k)}$ of the matrix represents the score that is outputted from the network having parameter θ at the k-th word, for the i-th tag, for the sentence $[y]_1^K$. For the CRF layer, there is a state transition matrix as parameters $[A]_{(i,j)}$ to model the transition from i-th to j-th state for a pair of successive time-steps. The score of a sentence is then given by the sum of the network and the transition scores. For more details refer [5,8].

– **Model 2: Bi-LSTM/GRU - CRF Networks.** Analogous to the LSTM/GRU-CRF network, this particular model combines a bi-directional LSTM/GRU network with a CRF model to obtain a Bi-LSTM/GRU-CRF model shown in Fig. 7. Therefore, along with the past input features and sentence level tag information as used in a LSTM/GRU-CRF model, the model utilizes the future input features as well. The training algorithm for the Bi-LSTM/GRU-CRF model is shown in Algorithm 1. For more details of the algorithm, refer [5]. All the proposed models in this paper use a generic Stochastic Gradient Descent forward and backward training method.

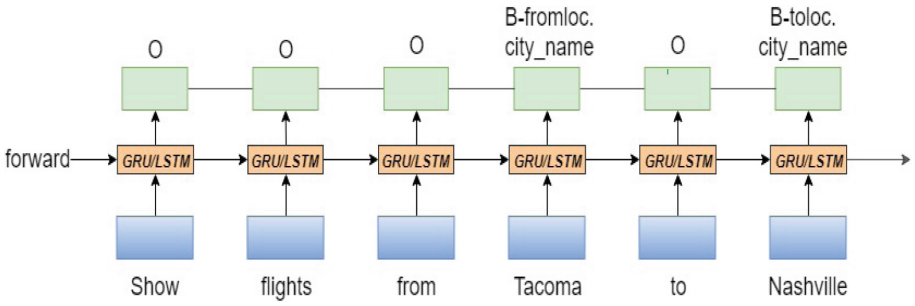

Fig. 6. A LSTM/GRU-CRF model

4 Experimentation, Results and Analysis

This section demonstrates the experimentation, results and analysis of all the proposed approaches. Number of utterances in training, validation and testing

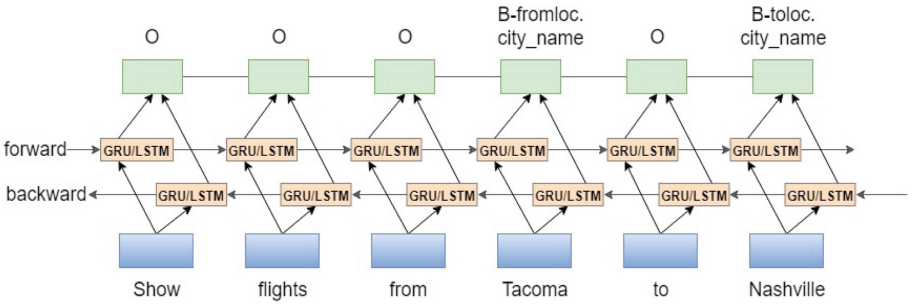

Fig. 7. A Bi-LSTM/GRU-CRF model

Algorithm 1. Bidirectional LSTM/GRU-CRF model training procedure

```
begin
    for each epoch do
        for each batch do
            1) bidirectional LSTM/GRU-CRF model forward pass:
                forward pass for forward state LSTM/GRU
                forward pass for backward state LSTM/GRU
            2) CRF layer forward and backward pass
            3) bidirectional LSTM/GRU-CRF model backward pass:
                backward pass for forward state LSTM/GRU
                backward pass for backward state LSTM/GRU
            4) update parameters
        end
    end
end
```

set for the benchmark ATIS [15] dataset are shown in Table 2. Since the ATIS dataset does not have a standard validation dataset, part of the training data has been used for the validation purpose.

4.1 Experimentation

For implementing the DNN models, Keras[4] has been used. In the input layer, all the unique words of the corpus are given some sequence numbers which are fed to the embedding layer. Pretrained GloVe [14] embedding trained on the CommonCrawl corpus of dimension 300 has been used to represent input words as word vectors. The resultant word embeddings from the input layer are fed to the LSTM/GRU layer for discourse modeling. Number of LSTM/GRU units in a layer is 100. A learning rate of 0.1 is used to train the models. For the baseline models, the number of units in the dense layer is equivalent to the number of

[4] https://keras.io/.

Table 2. No. of tokens and sentences in training, validation and testing sets of ATIS dataset

	Train set	Validation set	Test set
# Tokens	47604	8987	9198
# Utterances	4181	797	893

unique tags in the tag-set. Next, the softmax activation is used at the output layer and categorical crossentropy is used as the loss function.

Table 3. Results of all the baseline models

Models	Accuracy	F1-Score
LSTM	95.35	95.06
GRU	94.29	94.95
Bi-LSTM	96.86	96.63
Bi-GRU	95.79	95.56
CRF	72.80	68.91

4.2 Results and Analysis

Results of all the baseline models on the test set are presented in Table 3. It is quite evident from the table and as expected the bi-directional networks perform better since they can model both past and future dependencies. Results of all the proposed models are shown in Table 4. Therefore, the best performing model as seen from the table is that of a Bi-LSTM-CRF model which attains 2% and 26% increments over the corresponding Bi-LSTM and CRF baseline models, respectively in terms of accuracy. We have performed Welch's t-test [21] at 5% significance level and the corresponding results are shown in Table 5. This test signifies that the results produced by all our best performing models are statistically significant.

Table 4. Results of all the proposed models

Models	Accuracy	F1-Score
LSTM-CRF	97.09	96.97
GRU-CRF	96.89	96.38
Bi-LSTM-CRF	98.15	**97.94**
Bi-GRU-CRF	97.71	97.44

Table 5. p-values obtained by Welch's t-test comparing our best performing model with other models

Models	p-values
LSTM-CRF	2.13E−28
GRU-CRF	4.87E−33
Bi-GRU-CRF	3.55E−13

Table 6. Comparison of the proposed approach with the state-of-the-art

Models	F1-Score
RNNs (Mesnil et al. [12])	93.98
RNNs (Yao et al. [22])	96.60
R-CRF (Mesnil et al. [11])	96.46
s-CNN (Vu [19])	95.61
Bi-LSTM-CRF (Our Model)	**97.94**
Bi-GRU-CRF (Our Model)	**97.44**

4.3 Error Analysis

In order to analyze the weakness of the developed model, we have carried out a thorough error analysis of the proposed model. Since the number of unique slot labels in the ATIS corpus is 127, the representation of most of the tags are very less i.e. the dataset is skewed having lesser occurrences of most of the slot labels. This is one of the reasons for the errors. Example utterance such as "*which flights arrive in burbank from las vegas on* ***saturday april twenty third*** *in the afternoon*", here the words marked in bold are wrongly tagged as "*B-arrive_date.day_name*", "*B-arrive_date.month_name*", "*B-arrive_date.day_number*", "*Iarrive_date.day_number*". It should have been tagged as "*B-depart_date.day_name*", "*B-depart_date.month_name*", "*B-depart_date.day_number*", "*I-depart_date.day_number*", respectively. Similarly, "*find nonstop flights from salt lake city to new york on* ***saturday april ninth***", have been wrongly tagged as "*B-arrive_date.-day_name*", "*B-arrive_date.month_name*", "*B-arrive_date.day_number*" whereas it should have been tagged as "*B-depart_date.day_name*", "*B-depart_date.month_name*", "*B-depart_date.day_number*", respectively. Another such utterance "*does* ***tacoma airport*** *offer transportation from the airport to the downtown area*" is wrongly tagged as "*B-toloc.airport_name*", "*I-toloc.airport_name*" whereas it should have been tagged as "*B-airport_name*", "*I-airport_name*", respectively. Mostly, it was found that the errors occurred because the model was not able to distinguish between arrival and departure details.

Comparison with the state-of-the-art approaches. A comparative study has been carried out between our best performing proposed model against the state-of-the-art approaches shown in Table 6. It is evident from the table that

our best performing model, Bi-LSTM-CRF and Bi-GRU-CRF, outperformed various state-of-the-art approaches.

5 Conclusions and Future Work

In this paper, various model architectures are proposed for the task of slot-filling to capture the past and future dependencies of the input sentence along with the sentence level tag information. The proposed model outperformed various state-of-the-art approaches on the benchmark ATIS dataset.

In future, we aim to assess the proposed models on datasets belonging to varied domains. Also, we would like to extend our work to investigate different deep learning techniques to increase the accuracy of our model.

References

1. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. 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: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 210–215. IEEE (2012)
3. He, Y., Young, S.: A data-driven spoken language understanding system. In: 2003 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2003, pp. 583–588. IEEE (2003)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
6. Kalchbrenner, N., Blunsom, P.: Recurrent convolutional neural networks for discourse compositionality. arXiv preprint arXiv:1306.3584 (2013)
7. Khanpour, H., Guntakandla, N., Nielsen, R.: Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2012–2021 (2016)
8. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
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. Macherey, K., Och, F.J., Ney, H.: Natural language understanding using statistical machine translation. In: Seventh European Conference on Speech Communication and Technology (2001)
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. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: INTERSPEECH, pp. 3771–3775 (2013)

13. Moschitti, A., Riccardi, G., Raymond, C.: Spoken language understanding with kernels for syntactic/semantic structures. In: IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU, pp. 183–188. IEEE (2007)
14. 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)
15. Price, P.J.: Evaluation of spoken language systems: the ATIS domain. In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, 24–27 June 1990 (1990)
16. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Conference on Empirical Methods in Natural Language Processing (1996)
17. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding. In: Eighth Annual Conference of the International Speech Communication Association (2007)
18. Tur, G., Deng, L., Hakkani-Tür, D., He, X.: Towards deeper understanding: deep convex networks for semantic utterance classification. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5045–5048. IEEE (2012)
19. Vu, N.T.: Sequential convolutional neural networks for slot filling in spoken language understanding. arXiv preprint arXiv:1606.07783 (2016)
20. Wang, Y.Y., Acero, A., Mahajan, M., Lee, J.: Combining statistical and knowledge-based spoken language understanding in conditional models. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 882–889. Association for Computational Linguistics (2006)
21. Welch, B.L.: The generalization ofstudent's' problem when several different population variances are involved. Biometrika **34**(1/2), 28–35 (1947)
22. Yao, K., Zweig, G., Hwang, M.Y., Shi, Y., Yu, D.: Recurrent neural networks for language understanding. In: Interspeech, pp. 2524–2528 (2013)
23. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: IJCAI, pp. 2993–2999 (2016)