Neural Computing and Applications (2022) 34:13639–13648
https://doi.org/10.1007/s00521-022-07171-y

ORIGINAL ARTICLE

Understanding users' requirements precisely: a double Bi-LSTM-CRF joint model for detecting user's intentions and slot tags

Chunshan Li[1,2] · Yingli Zhou[2] · Guoqing Chao[2] · Dianhui Chu[2]

Received: 8 December 2021 / Accepted: 2 March 2022 / Published online: 31 March 2022

Abstract

Understanding users' requirements are essential to developing an effective AI service system, in which requirement expressions of users can be resolved into intent detection and slot filling tasks. In a lot of literature, the two tasks are normally considered as independent tasks and obtain satisfactory performance. Recently, many researchers have found that intent detection and slot filling can benefit each other since they always appear together in a sentence and may include shared information. Most of the existing joint models employ the structures of encoder and decoder and capture the cross-impact between two tasks by concatenation of hidden state information from two encoders, which ignore the dependencies among slot tags in specific intent. In this paper, we propose a novel Double-Bi-LSTM-CRF Model (DBLC), which can fit the dependency among hidden slot tags while considering the cross-impact between intent detection and slot filling. We also design and implement an intention chatbot on the tourism area, which can assist users to complete a travel plan through human-computer interaction. Extensive experiments show that our DBLC achieves state-of-the-art results on the benchmark ATIS, SNIPS, and multi-domain datasets.

Keywords Intent detection · Slot filling · LSTM · Conditional random fields

1 Introduction

The ability to understand users' requirements accurately in a conversation is essential to develop an effective AI service system, e.g., task-oriented order system, intelligent customer service system, and recommendation systems. For example, in the customer service of travel domain, "I want to book a taxi from Beijing Olympic Center to Changan Hotel today", an AI system should correctly understand that the user's intention is "booking taxi". Meanwhile, the AI system also should know "Beijing Olympic Center","Changan Hotel" and "today" are the departure, destination, and date of the travel respectively. As shown in Table 1, these precise requirements are typically represented through semantic format tags, and extracting such format information tags involve two tasks: intent detection and slot filling.

In the early time, the two tasks are normally considered as independent tasks. The intent detection task can be considered as utterance classification problem [1–4], which can be addressed by classical machine learnings, such as support vector machines [5] and boosting-based classifiers [6]. Latter, deep neural network-based intent detection attracts the attention of researchers because of the state-of-art performance, e.g., Convolution Neural Networks [7], Recurrent Neural Networks [8], Long Short-Term Memory Network [9], Gated Recurrent Unit [10], Attention Mechanism [11] and Capsule Networks [12]. At the same time, the slot filling task can be formulated as a sequence tagging problem. And the conditional random fields (CRFs) [13]

✉ Chunshan Li
lics@hit.edu.cn

✉ Dianhui Chu
chudh@hit.edu.cn

Yingli Zhou
zhouyl@hit.edu.cn

Guoqing Chao
chaoqq@hit.edu.cn

[1] State Key Laboratory of Communication Content Cognition, People's Daily Online, Beijing 100733, China

[2] Department of Computer Science, Harbin Institute of Technology, Weihai 264209, Shandong, China

Table 1 Semantic format of a utterance

Utterance	Book	Taxi	Form	Beijing	Olympic	Center
True slot	O	O	O	B-dept	I-dept	I-dept
Mis-predicting slot	O	O	O	B-dept	I-dept	I-dept
Utterance	to	Changan	Hotel	today		
True slot	O	B-des	I-des	B-date		
Mis-predicting slot	O	**O**	**B-des**	B-date		
Intent	Book_Taxi					
Domain	Travel					

Bold indicates the contents of these cells are important

and recurrent neural networks (RNN) [14] are the common approaches.

In recent years, many researchers have found that intent detection and slot filling can benefit each other [15, 16], since they always appear together in a sentence and may include shared information. There are two common routes for the joint model. The first one is that one unified encoder is used to read input utterances, and two decoders generate sequential intent and semantic tags, respectively. The second way is using two encoders and two decoders, and the cross-impact between two tasks can be captured by concatenation of hidden state information from two encoders. Wang et al. [17] propose a Bi-model based RNN structure to utilize the cross-impact between intent detection and slot filling tasks. The Bi-model builds two task networks to detect intents and semantic slots. The relationship between two tasks can be captured by the concatenation of hidden states of two task networks. Although Bi-model has achieved good performance, it cannot capture the dependencies and constraints among slot tags in specific intent. As shown in Table 1, when existing models make a prediction, these slot tags with the highest score are selected as the final labels. But in fact, the label with the highest score is not the most appropriate label necessarily. If two-slot words can be semantically recognized as an object (Beijing Olympic Center Changan Hotel), existing models always make a mis-predicting.

In this paper, we proposed a novel Double-Bi-LSTM-CRF Model (DBLC), which can fit the dependency among slot tags while considering the cross-impact between intent detection and slot filling. Specifically, the DBLC model constructs two networks to handle the intent detection task and slot filling task separately. Each task network employs a Bi-LSTM as the encoder and an LSTM as the decoder. The cross-impact between two tasks can be captured by the concatenation of hidden states from two encoders. The conditional random fields (CRFs) structure can be adopted to learn the dependency and constraints among different slot tags. We also employ the way of asynchronous training to infer DBLC, which trains two task networks with varying functions of cost.

The main contributions of this paper are as follows:

- We propose the DBLC model to construct two cooperative task networks to handle the intent detection task and slot filling task together. The DBLC model can analyze and obtain more accurate users' requirements from their utterances.
- We use conditional random fields (CRFs) structure to fit the dependency among slot tags while considering the cross-impact between intent detection and slot filling.
- We adopt the asynchronous training method to infer the DBLC model, which can keep the independence of the two tasks and capture more useful information while training two task networks with different cost functions.
- Experiments conducted on three real-world data sets show that the proposed DBLC model is effective and outperforms the state-of-the-art methods.
- We design and implement an intention chatbot, which can assist users to complete a travel plan through human-computer interaction.

The rest of the paper is organized as follows. In Sect. 2 we introduce the brief related works for the intent detection, slot filling, and joint model. In Sect. 3, we mainly discuss more details about Double-Bi-LSTM-CRF Model, including model structure and training process. In Sect. 4, we design the experiments to evaluate DBLC and analyze the experimental results from different perspectives. In Sect. 5, We design and implement an intention chatbot, which can illustrate the performance of our DBLC model on tourism domain. Finally, we draw conclusions and plan our further works in Sect. 6.

2 Related work

This section provides a brief overview of several related works which are most relevant to the proposed method. There are intent detection tasks, slot filling tasks, and joint models for both.

2.1 Intent detection

Many researchers consider the intent detection tasks as the first step to analyze the user's requirements. They employ a great number of classical machine learning or deep learning-based methods to deal with. In the early time of the study, support vector machines [5], or boosting-based classifiers [6] are common methods. In recent years, several Deep Learning-based methods have been explored because of the state-of-art performance, such as word embedding, Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory Network (LSTM), Gated Recurrent Unit (GRU), Attention Mechanism and Capsule Networks. Kim et al. [18] use word embedding as the initial representation of words and build LSTM for intent detection. Hashemi et al. [7] employ CNN to extract semantic features to identify the user's query intents in a dialog system. Bhargava et al. [8] investigate the intent detection work using the RNN+CRF model. Ravuri et al. [10] propose a word encoding based on character n-grams and employ LSTM and GRU to understand the user's intents. Lin et al. [11] introduce a self-attention mechanism to weight intent representation. Xia et al. [12] proposed an intent capsule model to discriminate emerging intents via knowledge transfer from existing intents. Although these methods conducted good results, they just consider intent detection as a text classification problem. Many specific intention points still cannot be revealed.

2.2 Slot filling

The slot filling tasks are seen as the second step to analyze the user's requirements, which can determine the specific points of intention. The slot filling tasks can be addressed by supervised sequence labeling methods, e.g., Maximum Entropy Markov Models (MEMMs), Condition Random Field (CRF), Recurrent Neural Networks (RNNs), or a combination of these models [19]. Guo et al. [20] proposed RecNN model, which provides a better mechanism for incorporating both discrete syntactic structure and continuous-space word and phrase representations. Liu et al. [21] introduce label dependencies in RNN model training by feeding previous output labels to the current sequence state. Xu et al. [22] combine the convolutional neural networks and triangular CRF model to exploit dependencies between the intent label and the slot sequence. Kurata et al. [23] proposed encoder-labeler LSTM to encode input sequence into a fixed-length vector and predict the label sequence. Deoras et al. [24] employ deep belief networks (DBN) to tag the semantic sequence. Sukhbaatar et al. [25] introduce a neural network with a recurrent attention model for language modeling and slot tagging.

2.3 Joint model for intent detection and slot filling

Despite intent detection and slot filling having achieved excellent performance independently, many researchers still consider that they can benefit each other since the two tasks always appear together in a sentence and may include shared information. Early classic work is CNN+Tri-CRF model [22], which uses convolutional neural networks as a shared encoder for both tasks and then CRF model to exploit dependencies between both tasks. Guo et al. [20] use the node representation on the syntactic tree of the utterance as a shared encoder among intent detection and slot filling. Zhang et al. [26, 27] share a bi-GRU encoder and a joint loss function between two tasks. Liu et al. [28] employ bi-directional LSTM as a shared encoder and two different decoders for joint tasks. They also use the attention mechanism to learn relations between slot labels in the decoder and words in the encoder. Goo et al. [29] considering that slot and intent have a strong relationship and use a slotted gate to learn the relationship. Recently, Wang et al. [17] proposed a bi-LSTM structure-based joint model to learn the dependencies between intent detection and slot filling. They adopt two independent bi-LSTM to read the input sentences, then the encoded information can be shared with other tasks. The Bi-model [17] use asynchronous training with two different loss function and achieve excellent performance. Qin et al. [30, 31] proposed two self-attentive-based models that can produce better context-aware representations to guide the slot filling task and detect intent at the word level. Most recently, large-scale pre-trained language models, such as BERT [32], ERNIE [33] and XLNet [34] have shown great improvements in many NLP service system. Chen et al. [35] investigate a transformer mechanism for joint tasks by fine-tuning a pre-trained BERT model, in which the BERT model employs a context-dependent sentence representation and builds a multi-layer bidirectional transformer encoder to receive the input. Zhang et al. [36] employ a multi-task learning model-based transformer encoder-decoder framework to conduct joint training for joint tasks. The model encodes the input sequence as context representations using bidirectional encoder representation from transformers and implements two separate decoders to handle the intent detection and slot filling.

2.4 Chatbots

Intent Detection and Slot Filling are the first steps to understanding users' requirements. If an AI system wants to mimic human conversation and serve users, it should build a Human-Computer Interaction program [37], which is an intent chatbot. Most of the early chat robots are conversation robots [38]. Personal voice assistants like Siri, Xiaodu, or Alexa are typical conversation chatbots, which can talk to the user like another human being. The key factor of a conversation-based chatbot is understanding the context of sentences and responding correctly to the conversation that it met. Although conversation chatbots have attracted great attention in academia and industry with the emergence of commercial, personal assistants, they are incapable of holding a multi-turn conversation to perform a specific task such as booking a restaurant or serving users. Papaioannou et al. [39] employ Reinforcement Learning to create an effective approach to combining chat and task-based dialogue for multimodal systems and deal with unforeseen user input. Li et al. [40] implement a task-oriented chatbot as a speaking teaching assistant, which allows users to continuously improve their language fluency in terms of speaking ability by simulating conversational situational exercises. Focusing on a particular domain or task, these chatbots seem to hold tremendous promise for providing users with quick and convenient service responding specifically to their questions.

3 Methodology

In this section, we will first formulate the joint intent detection and slot filling tasks. Then, we present the novel Double-Bi-LSTM-CRF Model (DBLC), which can fit the dependency among hidden slot tags while considering the cross-impact between intent detection and slot filling.

3.1 Problem formulation

Intent detection and slot filling can be formulated as a joint prediction task as follows: Given an input sequence $x = \{x_1, \ldots, x_n\}$, where each $x_i \in x$ represents the input vector of the ith word, The x will predict an intent objective y_{intent} and a sequence of tags $y = \{y_1, \ldots, y_n\}$. The y_{intent} indicates intent of input sentence and each label $y_i \in y$ is slot tag or Non-tag (not semantic slot) which corresponds to x_i. The formulation of our joint predicting task takes into account two factors to generate state of art prediction. One is the correlations between intent detection task and slot filling task. And another is the correlations among neighboring tags (semantic slots).

3.2 Joint model

Figure 1 shows the network structure of the proposed joint model, in which two bidirectional LSTMs (Bi-LSTMs) work together to predict intents and semantic slots. The top part of the network is used for intent detection. In the top part, a Bi-LSTM reads and encodes the input sentence $x = \{x_1, \ldots, x_n\}$ not only forwardly but also backwardly, and generates two forward and backward hidden states as $\overrightarrow{\mathrm{h}_t}$ and $\overleftarrow{\mathrm{h}_t}$. A concatenation of $\overrightarrow{\mathrm{h}_t}$ and $\overleftarrow{\mathrm{h}_t}$ can generate a final hidden state $h_t = \{\overrightarrow{\mathrm{h}_t}, \overleftarrow{\mathrm{h}_t}\}$ at time step t. Then another LSTM layer can be employed as the decoder, which adopts N vs. 1 strategy and predicts the intent of the input sequence. The bottom part of a network is used for slot filling, which has a similar structure as the top part and utilizes N vs. N strategy to identify the slot tags. In addition, the bottom network adds a CRF layer to capture the correlations among neighboring tags in the sequence, which improves the performance of the slot filling task obviously.

Specifically, we define that the bidirectional LSTM applies $f_i(.)$ to generate a sequence of hidden states $(h_1^i, h_2^i, \ldots, h_n^i)$, where $i = 1$ corresponds the top part network (intent detection task) and $i = 2$ is for the bottom part network (slot filling task).

In the intent detection task, the $f_1(.)$ can read the sentences word-by-word and generate an hidden representation h_t^1. Then h_t^1 and h_t^2 (representation from slot filling network) can be used as input for the decoder $g_1(.)$. Finally, the intention y_{intent}^1 can be predicted by LSTM $g_1(.)$. Let s_t^1 be the state of $g_1(.)$ at time step t:

$$\mathrm{s}_t^1 = \omega(s_{t-1}^1, h_{t-1}^1, h_{t-1}^2) \tag{1}$$

$$y_{\text{intent}}^1 = \underset{\hat{y}_n^1}{\operatorname{argmax}}\, P(\hat{y}_n^1 \| s_{n-1}^1, h_{n-1}^1, h_{n-1}^2) \tag{2}$$

where $\hat{y}_n^1$ includes the predicted probabilities for all intent labels at the last time step n.

To fill the semantic slot, a composite network structure is constructed with a bidirectional LSTM $f_2(.)$, a LSTM $g_2(.)$ and CRF network $c_2(.)$. Similarly, $f_2(.)$ can receive the input sequence. The $g_2(.)$ can generate a intermediate label sequence $l^2\cdot$. Finally, the slot tag sequence y_t^2 can be identified by CRF $c_2(.)$ at time step t:

$$\mathrm{s}_t^2 = \varphi(h_{t-1}^1, h_{t-1}^2, s_{t-1}^2, y_{t-1}^2) \tag{3}$$

$$l_t^2\cdot = \underset{l_t^2\cdot}{\operatorname{argmax}}\, P(\hat{l}_t^2 \cdot \| h_t^1, h_t^2, s_{t-1}^2, l_{t-1}^2\cdot) \tag{4}$$

$$y_t^2 = \underset{l_t^2}{\operatorname{argmax}} \sum_{i=1}^{N} \ln P(l_{t,i}^2 \| l_{t,i-1}) \tag{5}$$

3.3 Training procedure

To train the proposed model, we employ the way of asynchronous training, which trains two task networks with

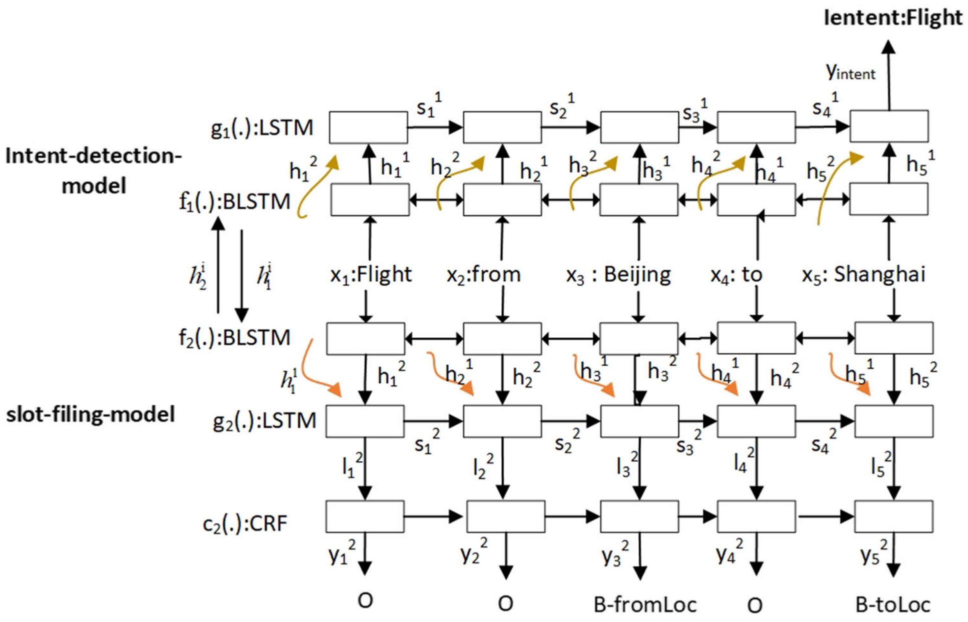


Fig. 1 Network structure of the double Bi-LSTM-CRF model

different cost functions. The reason for adopting separate cost functions is obvious. Because intent detection and slot filling are different NLU tasks, using the same cost function for both two networks will cause the differences between the tasks to be ignored, and much useful information may not be captured. Let L_1 be the loss function for the intent detection network and L_2 be the loss function for the slot filling network. And they can be defined as follows:

$$L_1 \triangleq -\sum_{i=1}^{k} \hat{y}_{\text{intent}}^{1,i} \log(y_{intent}^{1,i}) \tag{6}$$

$$\sum_{i=1}^{m} \hat{y}_j^{2,i} \log(y_j^{2,i}) \tag{7}$$

where k is the number of intention types, m is the number of slot-tag types, and n is the number of words in a sentence. In each training epoch, we split the whole data into batches and processed one batch at a time. Each batch includes several sentences, and the size of a batch can be set to 32, which means that the total length of sentences in a batch is less than 32. Next, we run Bi-LSTM $f_1(.)$ and $f_2(.)$ to read the input batch and generate a group of hidden representation h_t^1 and h_t^2 at time step t. Then, the intent detection network combine hidden states s_t^1 with h_t to predict intent label $\hat{y}_{\text{intent}}^1$. After that, we use L_1 to compute the cost of intent to detect the network, by which all parameters of the network can be trained. Furthermore, we employ a similar step to train a slot filling network. Finally, the CRF model is trained by maximizing the conditional log-likelihood:

$$\bar{\omega} = \arg\max_{\omega} \sum_{i=1}^{N} \ln p(y_i \| x_i, \omega) \tag{8}$$

where ω is model's parameters including the transition weights of CRF and the slot filling network.

4 Experimental analysis

4.1 Datasets

In this section, we conduct three real-world datasets to test the performance of the proposed Bi-LSTM-CRF model. The first one is the public ATIS (air travel information system) dataset, which is widely used in SLU research tasks. The ATIS contains sentences annotated with respect to intents and slots in the airline domain, in which there are 120 slot tags and 21 intent types. The second dataset is SNIPS (customintent-engines2) dataset, which is collected from the snips personal voice assistant. The snips data set contains 72 slot tags and 7 intent types. SNIPS dataset shows a more realistic scenario compared to the single domain of ATIS dataset. The details of ATIS and SNIPS datasets can be shown in the left part of Table 2. The third dataset is collected for three domains: food, home, and movie. Each domain contains three intents, and there are 15 slot tags in the food domain, 16 slot tags in the home domain, 14 slot tags in the movie domain. The details of the multi-domain dataset can be shown in the right part of Table 2, and the split is 70% for training, 10% for development, and 20% for the testing.

Table 2 Details of three real-world datasets

	ATIS	SNIPS	Movie	Food	Home
Vocabulary size	722	11,241	121	125	127
Slots	120	72	14	15	16
Intent	21	7	3	73	3
Training set size	4478	13,084	685	688	482
Development set size	500	700	98	98	69
Testing set size	893	700	196	197	138

4.2 Experimental setup

We mainly compare with the six recently introduced SLU models [17, 20, 22]. These models can simultaneously handle intent classification and slot filling. They all produced state-of-the-art results in the literature. For the setting of experimental parameters, we choose 200 as the size of the LSTM and Bi-LSTM networks. The number of hidden layers is set to 2. The size of the word embedding is 300, which are initialized randomly at the beginning of the experiment. The details of the hyper-parameters set can be shown in Table 3. As the report in literature [4, 17], we use accuracy to estimate the intent detection task and F1-Score to measure the slot filling task.

4.3 Experimental on ATIS and snips dataset

Our first experiment was conducted on the ATIS and Snips benchmark datasets. A detailed performance result is shown in Table 4. Compared with current state-of-the-art algorithms, the proposed model achieved the best performance on both intent detection tasks and slot filling tasks. We observe 0.2–0.3% absolute improvement in slot filling task. The reason for this advantage is that the CRF structure can capture the correlations among neighboring slot tags in sentences, which improves the performance of slot filling tasks obviously. We also see that the proposed model has the same performance in the intent detection task compared with the Bi-model. The possible reason for this phenomenon is that both Bi-model and our DBLC model build the same network structure to detect users' intentions which may generate similar results. In addition, Bi-model and our DBLC model obtain performance of 98.76% and 96.99% on the accuracy of intention detection task, which is quite high and difficult to be further improved.

Table 3 Hyper-parameters setting

Parameter	Value
Initial_learning_rate	0.3
Dropout	0.2
Embedding_size	300
Min_batch	32
Hidden_size	200
Hidden_layer	2
Epoch_num	500

4.4 Experimental on multi-domain dataset

To show the performance of the proposed model, we further conduct the experiment on a multi-domain dataset. Table 5 demonstrates that Double-Bi-LSTM-CRF model outperforms all comparing methods on slot filling task, in which we obtain 2.2% absolute improvement in the food domain, 0.5% absolute improvement in the movie domain, and 0.4% absolute improvement in the home domain. It is obvious that the proposed model can learn more information through correlations among slot tags. Firstly, the slot filling task has been the greatest improvement in the food domain. The possible reason for this phenomenon is that there are not many exact rules for the naming of food. Some names of food are long and strange, the existing model may not learn the relationship among name words. On the contrary, our model can identify these slot tags accurately, which can significantly improve the effectiveness of the actual scene. Secondly, the improvements in the movie domain and home domain are not very big. Since the naming of entities in these two areas is more regular than the food domain, the DBLC model only can correct little mispredicting of the decoder.

5 Application of intention chatbot

In order to illustrate the performance of the proposed model in a real-life scene, we design and implement an intention chatbot, which can help service providers and platforms to understand what are the user's exact requests and what they should offer [41–43]? As shown in Fig. 2, the intention chatbot consists of four functional modules, including user interface, spoken language understanding, dialogue management, and response generation.

The workflow of the intention chatbot starts with the user interface module, e.g., an user's query, "I want to book a taxi from Beijing Olympic Center to Changan Hotel today". Then, the spoken language understanding module will receive and analyze the user's request to infer the user's intention and slot tags (intent:"book_taxi", slots[departure:"Beijing Olympic Center", destination:"Changan Hotel", date:"today"]). After the chatbot understands the best request, it must determine how to do next. The dialogue management module will perform the requested actions or retrieve the knowledge base to find

Table 4 Performance of different models on ATIS and snips datasets

Model	ATIS Dataset		SNIPS Dataset	
	Slot(F1)	Intent(acc)	Slot(F1)	Intent(acc)
Recursive NN [20]	93.36	95.40	90.25	91.30
CNN-CRF [22]	94.30	95.80	91.45	92.15
Joint GRU Model [26]	95.19	96.10	92.31	93.34
Attention Encoder-Decoder NN [27]	95.42	97.30	92.79	94.53
Slot-Gated [29]	95.20	94.31	88.80	97.00
Bi-model [17]	95.7	98.76	93.89	96.99
DBLC-model	**96.13**	**98.76**	**94.11**	**96.99**

Bold indicates the contents of these cells are important

Table 5 Performance of different models on multi-domain dataset

Model	Movie		Food	
	Slot(F1)	Intent(acc)	Slot(F1)	Intent(acc)
Attention encoder-decoder NN [27]	92.1	92.86	92.3	98.48
Bi-model [17]	93.3	94.89	93.6	98.48
DBLC-model	**93.8**	**94.89**	**95.8**	**98.48**
Model	Home			
	Slot(F1)	Intent(acc)		
Attention encoder-decoder NN [27]	96.5	97.83		
Bi-model [17]	97.8	98.52		
DBLC-model	**98.2**	**98.55**		

Bold indicates the contents of these cells are important

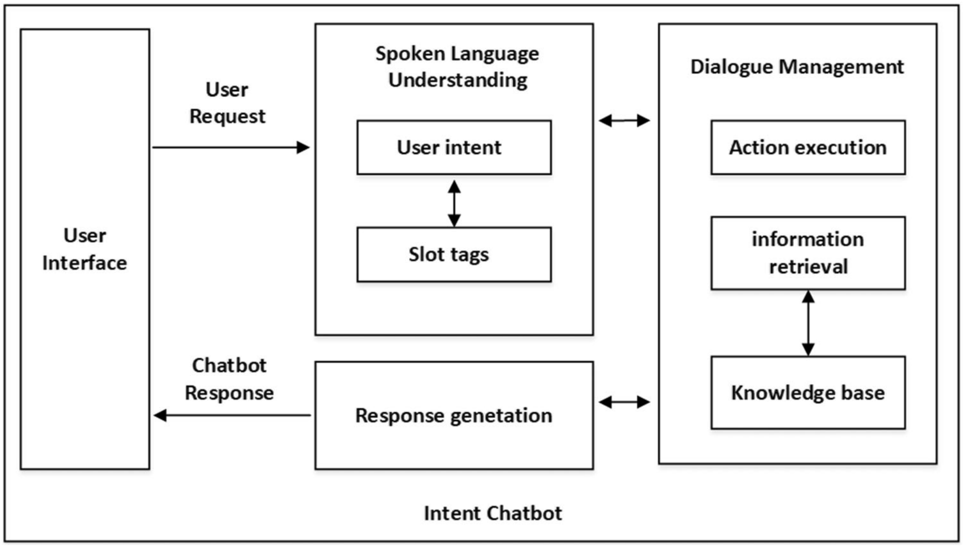
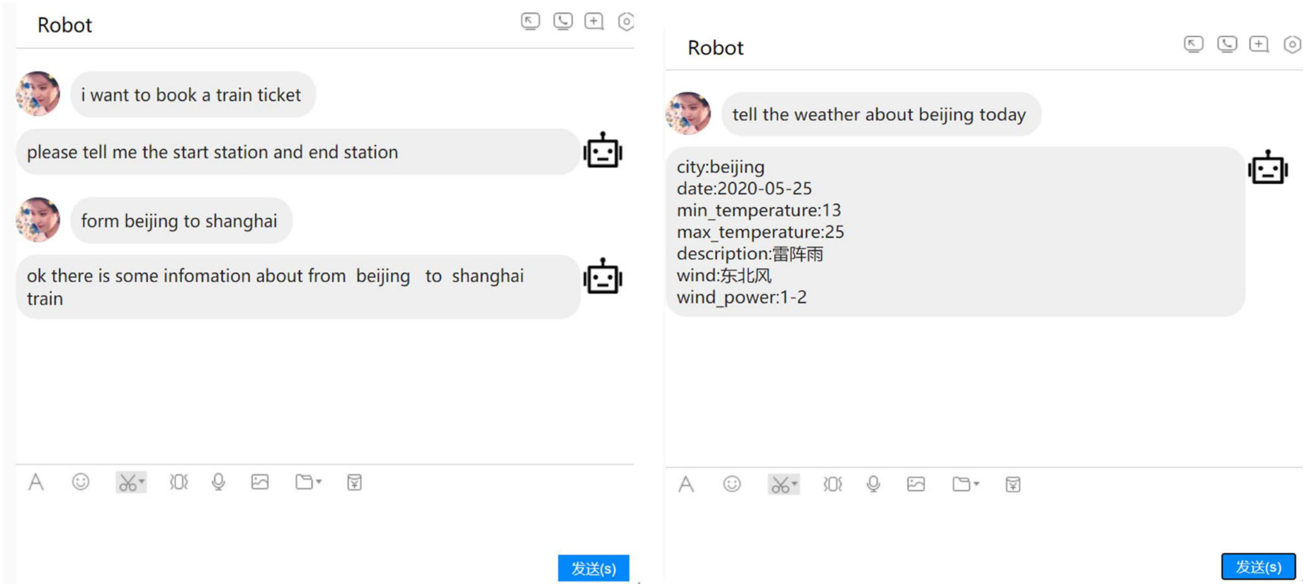


Fig. 2 Architecture of the intention chatbot

information that is appropriate to answer. Finally, the response generation module will prepare a natural language human-like sentence for the user.

The work scene of the intention chatbot concentrates on the tourism area which includes the daily chatting function, destination query function, travel reservation function as well as news query function. Through the chatbot, users



i want to book a train ticket

please tell me the start station and end station

form beijing to shanghai

ok there is some infomation about from beijing to shanghai train

(a) Travel reservation

tell the weather about beijing today

city:beijing
date:2020-05-25
min_temperature:13
max_temperature:25
description:雷阵雨
wind:东北风
wind_power:1-2

(b) Destination query

Fig. 3 Interface of chatbot

can make a travel plan. Specifically, the daily chatting function realizes the parsing and reply of basic greetings. The destination query function can offer the weather conditions in the next three days in major municipal cities around the world as well as the sunset and sunrise time. The travel reservation function can help users book train or plane tickets. The news query function supports four different theme news (Sports, fashion, science and technology, education), which are related to tourism destinations. Figure 3 shows the interface of chatbot. Users can enter their utterances from the input box. Then, the chatbot can parse users' intentions and talk to the user like another human being.

6 Conclusions and future works

In this paper, we propose a novel Double-Bi-LSTM-CRF (DBLC) Model that can identify the user's requirements accurately by way of resolving users' utterances into intentions and semantic tag sequences. To be specific, The DBLC model builds two networks to handle the intent detection task and slot filling task separately. Each task network employs a Bi-LSTM as an encoder and an LSTM as a decoder. The correlations between the two tasks can be captured by concatenating hidden states from two encoders. A CRF structure can be used to learn the dependency among slot tag sequences. Asynchronous training is employed to infer DBLC, which can keep the learning independence of the two networks and capture more useful information while training two task networks with different cost functions. Two experiments are conducted on three real-world datasets. One is performed on the ATIS and SNIPS benchmark datasets to demonstrate the state-of-the-art performance of the DBLC. The other experiment is tested on multi-domain datasets to show the performance of DBLC in a real-world scenario. We design and implement an intention chatbot, which can help service providers and platforms understand the user's exact requests are? Through the intention chatbot, users can complete a travel plan.

Although the proposed DBLC model can capture some significant information, such as cross-impact between intent detection and slot filling, dependency among slot tags, and so on, it still ignores some knowledge hidden in the users' intention. In fact, in a conversation, the intentions expressed by users are usually multiple and include the relationship with each other. In future, we plan to optimize our proposed method from different perspectives, e.g., multi-intention identification and relational network-based intention identification. We also will continue to implement a chatbot to recognize the user's requirements for home-based care.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (No. 61902090, 61772159, 61832004), the Natural Science Foundation of Shandong Province (No. ZR2020KF019) and by State Key Laboratory of Communication Content Cognition, People's Daily Online,(No. 2).

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. They have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Dopierre T, Gravier C, Subercaze J, Logerais W (2020) Few-shot pseudo-labeling for intent detection. In: Proceedings of the 28th international conference on computational linguistics, pp 4993–5003
2. Yan G, Fan, L, Li Q, Liu H, Zhang X, Wu X-M, Lam AY (2020) Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 1050–1060
3. Liu J, Li Y, Lin M (2019) Review of intent detection methods in the human-machine dialogue system. J Phys Conf Ser 1267(1):012059
4. Niu P, Chen Z, Song M et al (2019) A novel bi-directional interrelated model for joint intent detection and slot filling. arXiv preprint arXiv:1907.00390,
5. Chelba C, Mahajan M, Acero A (2003) Speech utterance classification. In: 2003 IEEE international conference on acoustics, speech, and signal processing, 2003. Proceedings.(ICASSP'03)., vol 1. IEEE, pp I–I
6. Schapire RE, Singer Y (2000) Boostexter: a boosting-based system for text categorization. Mach Learn 39(2–3):135–168
7. Hashemi H. B, Asiaee A, Kraft R (2016) Query intent detection using convolutional neural networks. In: International conference on web search and data mining, workshop on query understanding
8. Bhargava A, Celikyilmaz A, Hakkani-Tür D, Sarikaya R (2013) Easy contextual intent prediction and slot detection. In: IEEE international conference on acoustics, speech and signal processing 2013. IEEE, pp 8337–8341
9. Kapočiūtė-Dzikienė J (2020) Intent detection-based lithuanian chatbot created via automatic dnn hyper-parameter optimization. Front Artif Intell Appl 328:95–102
10. Ravuri S, Stolcke A (2015) Recurrent neural network and lstm models for lexical utterance classification. In: Sixteenth annual conference of the international speech communication association
11. Lin Z, Feng M, Santos CNd, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130
12. Xia C, Zhang C, Yan X, Chang Y, Yu PS (2018) Zero-shot user intent detection via capsule neural networks. arXiv preprint arXiv:1809.00385
13. Tang H, Ji D, Zhou Q (2020) End-to-end masked graph-based crf for joint slot filling and intent detection. Neurocomputing 413:348–359
14. Adel H, Schütze H (2019) Type-aware convolutional neural networks for slot filling. J Artif Intell Res 66:297–339
15. Chen S, Yu S (2019) Wais: word attention for joint intent detection and slot filling. Proc AAAI Conf Artif Intell 33:9927–9928
16. Ni P, Li Y, Li G, Chang V (2020) Natural language understanding approaches based on joint task of intent detection and slot filling for iot voice interaction. Neural Comput Appl 1–18
17. Wang Y, Shen Y, Jin H(2018) A bi-model based rnn semantic frame parsing model for intent detection and slot filling. arXiv preprint arXiv:1812.10235
18. Kim J-K, Tur G, Celikyilmaz A, Cao B, Wang Y-Y (2016) Intent detection using semantically enriched word embeddings. In: 2016 IEEE spoken language technology workshop (SLT). IEEE 2016, pp 414–419
19. Mesnil G, Dauphin Y, Yao K, Bengio Y, Deng L, Hakkani-Tur D, He X, Heck L, Tur G, Yu D et al (2014) Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans Audio Speech Lang Process 23(3):530–539
20. Guo D, Tur G, Yih W-T, Zweig G (2014) Joint semantic utterance classification and slot filling with recursive neural networks. In:2014 IEEE spoken language technology workshop (SLT). IEEE 2014:554–559
21. Liu B, Lane I (2015) Recurrent neural network structured output prediction for spoken language understanding. In: Proc. NIPS workshop on machine learning for spoken language understanding and interactions
22. Xu P, Sarikaya R (2013) Convolutional neural network based triangular crf for joint intent detection and slot filling. In: 2013 IEEE workshop on automatic speech recognition and understanding. IEEE 2013, pp 78–83
23. Kurata G, Xiang B, Zhou B, Yu M (2016) Leveraging sentence-level information with encoder lstm for semantic slot filling. arXiv preprint arXiv:1601.01530
24. Deoras A, Sarikaya R (2013) Deep belief network based semantic taggers for spoken language understanding. In: Interspeech, pp 2713–2717
25. Sukhbaatar S, Weston J, Fergus R et al (2015) End-to-end memory networks. Adv Neural Inf Process Syst 28:2440–2448
26. Zhang X, Wang H (2016) A joint model of intent determination and slot filling for spoken language understanding. IJCAI 16:2993–2999
27. Liu B, Lane I (2016) Joint online spoken language understanding and language modeling with recurrent neural networks. arXiv preprint arXiv:1609.01462
28. Liu B, Lane I (2016) Attention-based recurrent neural network models for joint intent detection and slot filling. arXiv preprint arXiv:1609.01454
29. Goo C-W, Gao G, Hsu Y-K, Huo C-L, Chen T-C, Hsu K-W, Chen Y-N (2018) 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, vol 2 (Short Papers), pp 753–757
30. Qin L, Che W, Li Y, Wen H, Liu T (2019) A stack-propagation framework with token-level intent detection for spoken language understanding. arXiv preprint arXiv:1909.02188
31. Qin L, Ni M, Zhang Y, Che W (2020) Cosda-ml: multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. arXiv preprint arXiv:2006.06402
32. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1)
33. Sun Y, Wang S, Li Y, Feng S, Tian H, Wu H, Wang H (2020) Ernie 2.0: a continual pre-training framework for language understanding. In: Proceedings of the AAAI conference on artificial intelligence, vol 34(05), pp 8968–8975
34. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems, vol 32
35. Chen Q, Zhuo Z, Wang W (2019) Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909

36. Zhang Z, Zhang Z, Chen H, Zhang Z (2019) A joint learning framework with bert for spoken language understanding. IEEE Access 7:168 849-168 858
37. Bansal H, Khan R (2018) A review paper on human computer interaction. Int J Adv Res Comput Sci Softw Eng 8:53–56
38. Murtarelli G, Gregory A, Romenti S (2021) A conversation-based perspective for shaping ethical human-machine interactions: the particular challenge of chatbots. J Bus Res 129:927–935
39. Papaioannou I, Dondrup C, Novikova J, Lemon O (2017) Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In: (2017) 26th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE 2017:593–598
40. Li K-C, Chang M, Wu K-H (2020) Developing a task-based dialogue system for English language learning. Educ Sci 10(11):306
41. Adamopoulou E, Moussiades L (2020) An overview of chatbot technology. In: IFIP international conference on artificial intelligence applications and innovations. Springer, pp 373–383
42. Adam M, Wessel M, Benlian A (2021) Ai-based chatbots in customer service and their effects on user compliance. Electron Mark 31:427–445
43. Eleni A, Lefteris M (2020) Chatbots: History, technology, and applications. Mach Learn Appl 2:100006