LEARNING SEMANTIC-MATCHING ATTENTION FOR MULTI-TURN SPOKEN LANGUAGE UNDERSTANDING

Ming JIANG[1], Kaiqiang CAO[2]*, Pengfei LI[3], Min ZHANG[4]

Spoken Language Understanding (SLU), one critical component of the task-oriented dialogue system, is responsible for parsing natural-language sentences into a machine-understandable representation with semantic frames. The user-defined semantic information is relatively complete and easy to be extracted in a single dialogue, but very short and rich in the several dialogues. As we all know, historical dialogue involves valuable information on processing the current sentences. However, how to effectively encode historical information, analyze internal dependencies and find out the relationship between historical and current sentences is still an unsolved challenging. This paper proposes a Tree LSTM combined with CNN to encode sentence information and calculate the degree of semantic relationship describing the connection of the historical and the current. The experiment shows the model we proposed successfully learns semantic matching attention from contextual encoding, which significantly improves accuracy on language understanding tasks.

Keywords: Spoken Language Understanding; semantic matching; intent detection

1. Introduction

SLU is aiming to form a semantic frame that captures the semantics of user utterances or queries. It typically involves two tasks: intent detection and slot filling[1]. These two tasks focus on predicting speaker’s intent and extracting semantic concepts as constraints for the natural language. Take a movie-related utterance as an example, *” find comedies by James Cameron”*, as shown in Fig 1. There are different slot labels for each word in the utterance, and a specific intent for the whole utterance. The most research on spoken language understanding is sentence analysis in single-round dialogue or multi-round dialogue. The sentence in the single-round dialogue is longer, which implicitly contains more constraints. Which can use traditional deep learning frameworks to achieve higher accuracy [2,3]. Compared with the single-round dialogue, the sentences in the multi-round dialogue are shorter, but with more complicated dependencies in each two rounds. Due to the demands on the context of multi-round dialogues, the present semantics can’t be accurately identified and clarified if just based on the current sentence.

[1] Prof., Computer Science and Technology, Hangzhou Dianzi University, China

[2]* M.E., Computer Technology, Hangzhou Dianzi University, China, corresponding author, e-mail: 17816869731@163.com

[3] Ph.D., Computer Science and Technology, Hangzhou Dianzi University, China

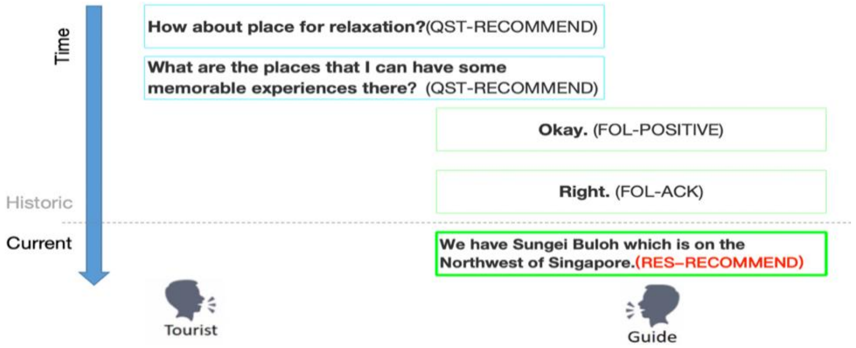
[4] Lect., Computer Science and Technology, Hangzhou Dianzi University, China

W	find	comedies	by	james	cameron
	↓	↓	↓	↓	↓
S	O	B-genre	O	B-dir	I-dir
I	find_movie				

Fig. 1. An example utterance with annotations of semantic slots in IOB format (S) and intent (I)

Thus, in order to address the issues of lacking historical information, [4] utilizes the memory network to store historical dialogue information, and calculates historical weight vector by summarizing the sentence information to understand the current sentence. However, this method only considers the historical sentence in the memory network, and completely neglects the relationship between historical and current sentences. [5,6,7] have proven the critical problems on spoken language understanding in multi-round dialogue are 1) how to find out the implicit relationship between the historical and the current, and 2) how to effectively extract weight vectors representing current sentences from historical information.

[8] proposes a brand-new method for extracting a feature vector based on the time dimension, whose process is to assign weight values to historical sentences according to the inverse distance between the historical sentence and the current one. However, in some cases, the time- and distance-based judgment performs wrong. For example, shown in Fig 2, spoken language understanding, receiving a sentence from tour guide, produces unexpected results if applying distance-based calculation to historical and current sentences.



How about place for relaxation?(QST-RECOMMEND)

What are the places that I can have some memorable experiences there? (QST-RECOMMEND)

Okay. (FOL-POSITIVE)

Right. (FOL-ACK)

We have Sungei Buloh which is on the Northwest of Singapore.(RES–RECOMMEND)

Fig. 2. An example of utterances with their semantic labels (speech acts combined with associated attributes) from DSTC 4. The semantic labels are italicized.

Based on the distance, "Right" has the greatest impact, but it has nothing to do with the current sentence. According to our manual check, the expected output should be "What are the places that I can have some memorable experiences there?", which looks like the most relevant one.

Taking the mentioned problems into account, we propose a sentence encoding model that utilizes optimized word vectors to extract local context features from a sentence using CNN and global semantic features from a sentence using Tree LSTM. Then by matching the semantics between the current sentence and the historical sentence, the model can automatically calculate the influence weight value of the historical sentence, and categorize them according to the two sides of the question and answer, then combine the historical sentence semantic representation and the corresponding sentence weight to get the role-based historical semantic weight vector. Finally, the historical semantic weight vectors of the two sides of the question and answer are combined to obtain the final historical weight vector, and the current sentence can be detected.

2. Related Work

Spoken Language Understanding has become a hot research topic in the task-oriented dialog system. In the last decade, machine learning and deep learning develop rapidly and turn into mature technologies gradually, which results in the relevant technologies of spoken language understanding have also achieved a great progress. In the single-round dialogue, traditional machine learning methods use n-gram vocabulary features as an input and put it into a pre-defined SVM algorithm for intent detection [9]. [10] proposes a tree model based on semantic dependency, which performs much better than rule-based approaches in terms of slot filling. In recent years, with the development of deep learning, intent detection and slot filling are considered as joint tasks. Through joint learning, the model can learn 'knowledge' from the shared features to reciprocally improve the accuracy of these two tasks. [2] uses BLSTM to exploit potentially shared features. [3] uses CNN to extract locally structural features of sentences, and jointly perform slot filling and intent detection. [11] proposes an attention LSTM model which focus on different parts of one sentence while predicting the slot value. [12] uses the slot-gated model to determine the relationship between intent and slot. [13] uses capsule neural networks to further strengthen the link between slot and intention.

In the multi-round dialogue, it has been proven that contextual information has a great positive influence on parsing the current sentence [5, 6]. Therefore, [4] stores historical sentence information into the memory network, and obtains the historical weight vector by summarizing them. [7] not only serializes historical sentences into a memory network to maintain time information, but also obtain historical vectors by a fully connected neural network.

Multi-turn conversations are usually carried out by two users. [14] proposes divide the historical sentence into question and answer sides and serializes them into the BLSTM to obtain two role history vectors that will be merged into one final history vector. Based on this paper, [8] proposes a method to assign influence weight values to historical sentences using the reciprocal distance away from the current sentence. and then combining historical sentence representations with weight values, put it into the corresponding role BLSTM to obtain the final historical vector. [15] proposes a universal time model by training three linear combinations of time decay functions (linear, convex, and concave) to dynamically assign the impact weight values to historical sentences. [16] proposes to utilize the user information of the current sentence to calculate the influence weight of the historical sentence.

Different from the [16]'s models, we are inspired by knowledge about semantic matching. While calculating the impact weight on the historical sentence, we not only take the distance of the current sentence into account, but also utilize semantic matching degree to calculate historical sentence from the perspective of semantic matching.

3. Model

This section describes a multi-turn conversational spoken language understanding model we propose. Its architecture diagram shown in Fig 3 consists of three functional components: 1) to extract sentence semantic information; 2) to extract historical semantic influence vector. 3) to combine historical semantic weight vectors to perform intent detection on the current sentence.

3.1 Sentence Semantic Information Extraction

The deep neural network model for sentence semantic extraction in this paper is shown in the bottom of Fig 3. In order to guarantee the semantics of sentence extraction to involve both local and global information, we introduced the CNN to extract the local features of the sentence and the Tree LSTM model to extract the semantic features of the sentence.

3.1.1 Word Vector Optimization Based On Attention

Using the pre-trained word vector model Glove[18], the sentence is converted into a word vector matrix(X={x_1, x_2,..., x_L}). In order to strengthen the correlation between each pair of non-continuous words, we first calculate the word-level attention mechanism before inputting it into the CNN.

we construct a word vector matrix as X $\in R^{L \times D}$, where L is the length of the sentence, D is the dimension of the word vector, and $X = [x_1, x_2, \dots x_L]$. The goal of the attention mechanism is to assign probability values to words with greater semantic relevance while calculating the current word X_i, to generate a corresponding context word vector g_i. The context word vector g_i is calculated by formula (1):

$$g_i = \sum_{j \neq i} \alpha_{i,j} . x_j \qquad (1)$$

where $a_{i,j}$represents the weight obtained by word semantic matching calculation, and it requires $a_{i,j} \geq 0$. The calculation is performed by the softmax function, and the weights of the corresponding probability values of the words are 1.

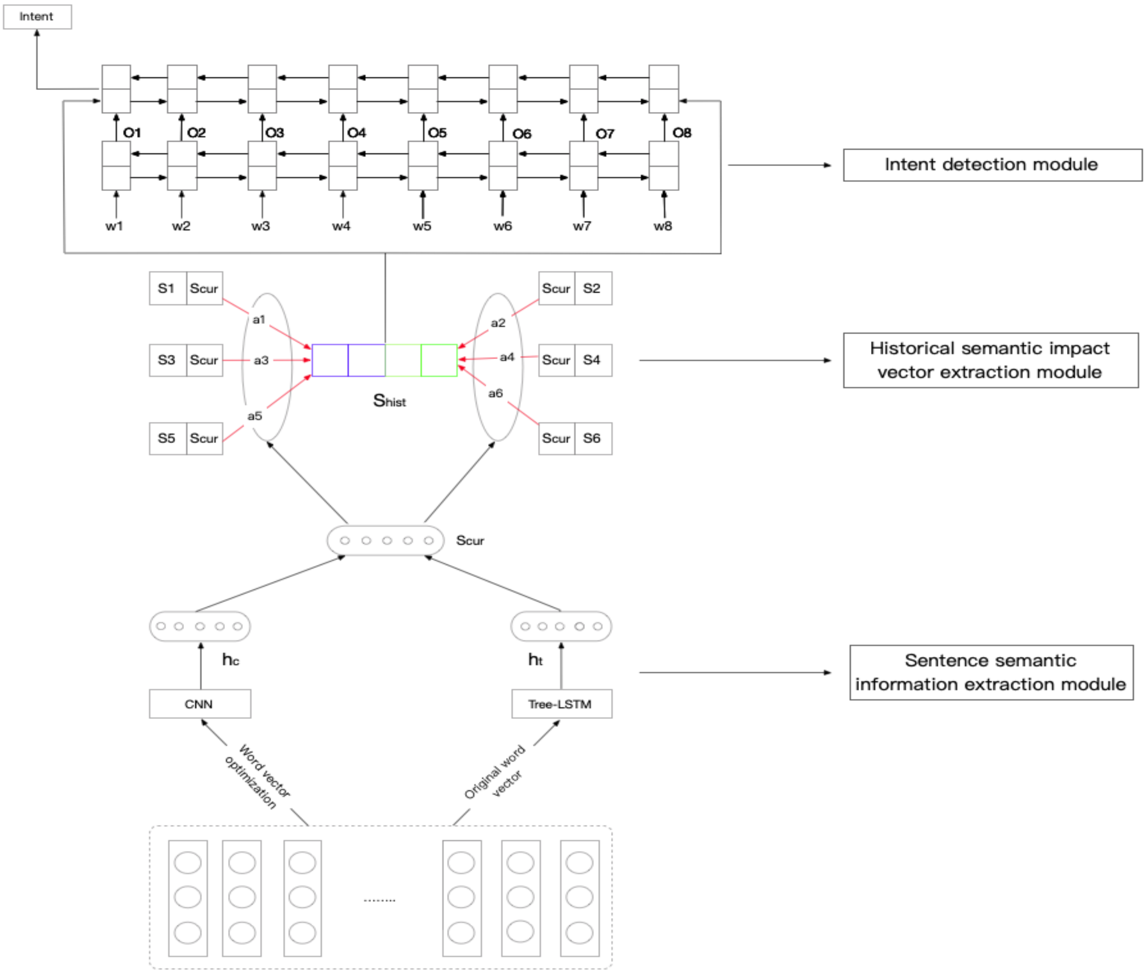


Fig. 3. The model proposed in this paper

The formula for calculating the weight of the attention mechanism is in formula (2):

$$\alpha_{i,j} = \frac{e^{score(x_i,x_j)}}{\sum_{j'=1,j'!=i}^{L} e^{score(x_i,x_{j'})}} \tag{2}$$

Where $score(x_i, x_j)$ is calculated based on the word vector representation distance between the remaining words in the sentence and the current word.

$$score(x_i\ ,\ x_j) = (x_i - \ x_j)^2 \tag{3}$$

After getting the context vector g_i corresponding to the word, combine it with the original word vector x_i to get a new word vector representation x_i' The new word vector matrix is $X'= [x_i^{'}, x_2^{'}, \dots x_L^{'}]$.

$$\mathrm{x_i^{'}} = \mathrm{x_i} \oplus \mathrm{g_i} \tag{4}$$

3.1.2 Local Feature Extraction

The CNN consists of an input layer, a convolutional layer, and a pooling layer.

The input layer considers the new word vector x'matrix as the input, $x' = [x_1^{'}, x_2^{'}, \dots x_L^{'}]$, where L is the length of the sentence and 2D is the new word vector dimension, word vector matrix dimensions are $X \in R^{L \times 2D}$.

The convolution layer extracts the local feature information of the sentence through the size of the variable window value, for example, the width of the context window is h and the weight matrix is $W \in R^{h \times 2D}$. Therefore, the feature c_i extracted by x_i' after the convolution operation is formula (5):

$$\mathrm{c_i} = \mathrm{ReLU}(\mathrm{W} \cdot \mathrm{x'_{i:i+h-1}} + \mathrm{b}) \tag{5}$$

Where we use Rectified Linear function as the activation function, $b \in R$ represents the corresponding bias term. After applying this filter to each word window $\{x'_{1:h}, x'_{2:h+1}, \dots x'_{L-h+1:L}\}$, it produces a series of feature maps:

$$\mathrm{c} = [\mathrm{c_1}, \mathrm{c_2}, \dots . \mathrm{c_{L-h+1}}] \tag{6}$$

Pooling layer receives a series of feature maps generated from the convolutional layer to get the sentence representation h_c operated by CNN.

3.1.3 Semantic Feature Extraction

After the sentence passes into the CNN, it can fully extract the local features of the sentence. Because the global semantic characteristics of sentences greatly impact on the accuracy metric of related tasks, we use a Tree LSTM to construct a semantic dependency tree based on the sentence semantics, and merge related nodes from the bottom up in the semantic dependency tree. As shown in Fig 4, The j-th node in the Tree LSTM includes: input gate i_j, output gate o_j, memory unit c_j and hidden unit h_j, the child nodes c_2 and c_3 both affects their common parent node c_1. For each child node k, unit j has a corresponding forget gate f_{jk}. as f_2 and f_3 shown in Fig 4. This makes the update of a node in Tree LSTM based on the information of multiple child nodes connected. Compared with the standard LSTM structure, the update of each node depends on the previous node in the sequence, and the update on the Tree LSTM depends on multiple child nodes in its semantic dependency tree.

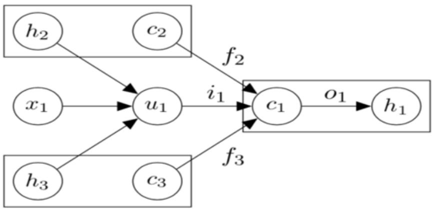


Fig. 4. Tree LSTM

The Tree LSTM model can use the output h_t of the root node as the final sentence semantic representation through continuous iterative training. Combining the sentence semantic representation of the Tree LSTM output with the sentence representation output of the CNN to obtain the final sentence S. The calculation method is formula (7):

$$\mathrm{S} = \mathrm{h_t} \oplus \mathrm{h_c} \tag{7}$$

Where h_t represents the sentence representation obtained through the Tree LSTM, $h_c$represents the sentence representation obtained by the CNN, and S represents the final sentence semantic representation obtained by concatenating the vector h_t and the vector h_c.

3.2 Historical Semantic Weight Vector

3.2.1 Semantic-Matching Attention

In this section, we will introduce the attention model based on semantic matching. In the previous chapter, we have obtained the representation S of the sentence. Which we use the first n current sentence as the historical sentence, 2) serialize the historical sentences into the section 3.1’s model to get the historical sentence matrix $S_{hist} = [S_1, S_2, \dots S_n]$, 3) put the current sentence into the section 3.1’s model to get the current sentence representation S_{cur}.

To calculate the impact weight value α_t of the t-th historical sentence, we

combine the historical sentence representation with the current sentence representation to obtain a new historical sentence matrix S'_{hist}=$[S'_1, S'_2, S'_3, \dots S'_n]$.

The merging process is formula (8):

$$S_t^{'} = S_t \oplus S_{cur} \tag{8}$$

The newly obtained historical sentence matrix is input to the MLP for data training to obtain the impact weight value of each historical sentence.

The calculation method is formula (9):

$$\alpha_t = \mathrm{softmax}\big(f(W_{att}^T \cdot S_t^{'} + b_{att})\big) \tag{9}$$

Where $W_{att}^T \in R^{n\times 2d}$ is a trainable weight transpose matrix. b_{att} is a trainable bias term. f is the activation function in the MLP network. We use the tanh activation function in MLP and softmax at the output for normalization.

3.2.2 Role-Level Historical Semantic Weight Vector

Multi-round dialogues are usually carried out by two or more speakers, where each speaker's speaking style and habits will directly impact on the understanding of the current sentence [14]. So in addition to sentence-level attention, we divide the historical sentences into two groups, which are the two sides of the question and answer separately, named as user A and user B temporarily. The historical sentence representation is combined with its probability value and input into the BLSTM of the corresponding role to obtain the role-based history vector, and the final historical sentence weight vector is composed of historical semantic vectors from both sides of the question and answer. The calculation method is formula (10):

$$V_{his} = BLSTM_{\mathrm{A}}(S_t, \alpha_t) + BLSTM_{\mathrm{B}}(S_t, \alpha_t) \tag{10}$$

3.3 Prediction

We combine the historical semantic weight vector obtained in the 3.2 section with the current sentence to perform intent detection. The calculation method is formula (11) and formula (12):

$$V_{cur} = BLSTM(S_{cur}, W_{his} \cdot V_{his}), \tag{11}$$

$$o = sigmoid(W_{LU} \cdot V_{cur}), \tag{12}$$

Where W_{his} is the weight matrix and V_{his} is the history vector obtained from training. V_{cur} is a context vector describing the current sentence. It comes from a BLSTM’s encoding by combining the current sentence with the historical semantic vector. o is the final intent distribution. Due this is a multi-label and multi-category classification, we use the sigmoid function at the end, where the user's final intention y will depend on whether the value of o is beyond the threshold.

4. Experiments

4.1 Setup

To evaluate the effectiveness of the model we proposed, we conduct the SLU experiments on human-to-human conversational data. The experiment used the open dialogue dataset DSTC4, which consists of 35 dialogue sessions on touristic information in Singapore collected from Skype calls between 3 tour guides and 35 tourists[19]. All recorded dialogues with the total length of 21 hours have been manually transcribed and annotated with speech acts and semantic labels at each turn level. We selected 14 dialogues as the training set, 6 dialogues as the testing set, and 15 dialogues as the validation set.

We chose the mini-batch *Adam*[20] as the optimizer with the batch size of 256 examples. The size of each hidden recurrent layer is 128. We used pre-trained 200-dimensional word embeddings Glove[18]. We only applied 30 training epochs without any early stop approach. The historical information range covers 10 sentences. We run each model 10 times, take an average as the final F1 score and calculate standard deviation of the model.

4.2 Model Comparison

In order to evaluate the multi-turn semantic-matching model that performs spoken language understanding, we designed an experiment to compare different models' performance on DSTC4 benchmark. As shown in Table 1, Sentence embedding indicates which model is used to encode sentences, Sentence attention indicates which criteria are used to assign weights to historical sentences, and use speaker role indicates whether to group historical sentences, that is, they are divided into tourist and tour guide. Use speaker indicator indicates whether to use the current user role information.

Table 1

The method used by different models

Model	**Sentence Embedding**	**Sentence Attention**	**Use Speaker Role**	**Use Speaker Indicator**
Speaker Role Modeling[14]	CNN	/	Y	N
Content-Aware Attention	CNN	Content-Aware	Y	N
Time-Aware Attention[8]	CNN	Time-Aware	Y	N
Universal Time-Aware Attention[15]	CNN	Universal Time- Distance-Decay-Aware	Y	N
Decay-Function-Free Time-Aware Attention With Speaker Indicator[16]	CNN	Decay-Function-Free-Aware	Y	Y
Decay-Function-Free-Content-and-Time-Aware Attention with Speaker Indicator	CNN	Decay-Function-Free-Content + Time-Aware	Y	Y
Semantic-Matching-Aware Attention	Tree LSTM + CNN	Semantic-Matching-Aware	Y	N
Semantic-Matching +Universal Time-Aware Attention	Tree LSTM + CNN	Semantic-Matching + Universal Time-Aware	Y	N

We design an experiment to compare models' accuracy on the benchmark DSTC4 SLU task. Sentence-Level means assigning weight values based only on the order of historical sentences, Role-Level means first grouping historical sentences based on users, and then assigning weight values. Its results shown in Table 2, illustrate the Semantic-Matching-Aware Attention model has higher accuracy than the Content-Aware Attention model in slot filling and intent detection, and the standard deviation of the model has also reached the smallest, indicating that the model is relatively stable. Therefore, based on the semantic matching degree between the current sentence and the historical sentence, assigning a weight value to the historical sentence is more accurate than calculating the sentence similarity based on the sentence content. The Semantic Matching + Universal Time-Aware attention model, whose process is to obtain historical semantic weight vectors based on semantic matching and universal time decay function, achieves the best result in the DSTC4's speaking language understanding task. The experiment proves the effectiveness of our model.

Table 2

The correctness of various models in SLU in DSTC4 dataset

Model	**Sentence-Level**				**Role-Level**			
	Slot Filling		**Intent**		**Slot Filling**		**Intent**	
	F1 Score	**STDEV.P**	**F1 Score**	**STDEV.P**	**F1 Score**	**STDEV.P**	**F1 Score**	**STDEV.P**
Speaker-Role Modeling	66.8	0.333	87.4	0.343	70.1	0.322	87.6	0.341
Content-Aware Attention	71.3	0.323	87.5	0.342	71.8	0.314	87.9	0.342
Time-Aware Attention	74.6	0.311	88.1	0.340	74.2	0.239	88.3	0.341
Universal Time-Aware Attention	74.22	0.236	88.45	0.340	74.12	0.235	88.52	0.339
Decay-Function-Free Time-Aware Attention With Speaker Indicator	75.59	0.231	89.67	0.337	76.1	0.23	89.68	0.334
Decay-Function-Free-Content-and-Time-Aware Attention with Speaker Indicator	76.1	0.227	89.66	0.334	76.50	0.225	89.67	0.332
Semantic-Matching-Aware Attention	75.62	0.22	89.98	0.330	75.84	0.218	90.04	0.328
Semantic Matching + Universal Time-Aware attention	75.90	0.213	90.07	0.325	**76.61**	**0.204**	**90.12**	**0.321**

Table 3

Comparison of different sentence encoding models

Model	**Sentence-Level**		**Role-Level**	
	F1 Score	**STDEV.P**	**F1 Score**	**STDEV.P**
CNN	73.60	0.245	73.81	0.240
Attention CNN	73.75	0.238	73.92	0.234
LSTM	74.61	0.233	74.80	0.233
BiLSTM	74.79	0.232	74.95	0.231

Tree-LSTM	75.23	0.228	75.34	0.223
LSTM+Attention CNN	74.92	0.227	75.06	0.225
BiLSTM+Attention CNN	75.13	0.225	75.27	0.224
Tree-LSTM+Attention CNN	75.62	0.22	**75.84**	**0.218**

4.3 Comparison Of Sentence Encoding Models

The method mentioned in this paper uses a Tree LSTM model in the sentence semantic information extraction module. In NLP, the popular models for extracting sentence serialization information are BLSTM. Herein, we compared different sentence encoding models, shown in Table 3, which are separately traditional neural network models to encode sentences, attention-based word-level CNN, LSTM-encoded sentence vector, BLSTM model and Tree LSTM model are merged respectively. Then, according to the remaining steps proposed in this paper, the weight distribution of the semantic matching degree between the historical sentence and the current sentence is performed to obtain the historical semantic influence vector, and the current sentence is used to intent detection. Compared with the traditional neural network model, the Tree LSTM is created based on the semantic tree. So Tree LSTM model can fully extract sentence semantics, and the Tree LSTM model is more stable than other models.

5. Conclusions

As for the spoken language understanding in multi-turns of dialogue, this paper proposes a method to calculate the weight of historical sentences based on semantic matching, and turn the question into a single-round retrieval dialogue. We also propose an attention-based word-level CNN model and apply it into Tree LSTM to fully extract sentence semantic information. Our method calculates the weighted influence value of the historical sentence by splitting the historical sentence semantic representation and the current sentence semantic representation into several pairs. The historical semantic weight vector is finally obtained by combining the historical sentence semantic representation and the corresponding weight value. Combine this vector to perform intent detection on the current sentence. Experiments show that this model we proposed can improve the accuracy of the intention prediction of the current sentence by 0.11% on the benchmark data set DSTC4.

Acknowledgements

This work is supported by the Zhejiang Provincial Technical Plan Project(No. 2020C03105).

REFERENCES

[1] *Y.-N Chen and J. Gao* Open-Domain Neural Dialogue Systems, in Proceedings of the IJCNLP 2017, Tutorial Abstracts.

[2] *Y. Wang, Y. Shen, and H. Jin.* A bi-model based rnn semantic frame parsing model for intent detection and slot filling, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2, volume 2, pages 309–314.

[3] *P. Xu, and R. Sarikaya.* Convolutional neural network based triangular CRF for joint intent detection and slot filling, in Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 78-83, 2013.

[4] *Y.-N Chen, D. H. Tur, G. Tur, J. Gao, and D. Li*. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding, in Proceedings of The 17th Annual Meeting of the International Speech Communication Association.2016

[5] *A. Bhargava, A. Celikyilmaz, D. Hakkani-Tur, and R. Sarikaya,* Easy contextual intent prediction and slot detection, in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2013, pp. 8337–8341.

[6] *P. Xu and R. Sarikaya.* Contextual domain classification in spoken language understanding systems using recurrent neural network, in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 136– 140.

[7] *A. Bapna, G. Tur, D. Hakkani-Tur, and L. Heck.* Sequential dialogue context modeling for spoken language understanding, in Proc.of SIGDIAL.2017.

[8] *P.-C Chen, T.-C Chi, S.-Y Su, and Y.-N Chen.* Dynamic time-aware attention to speaker roles and contexts for spoken language understanding, in Proceedings of ASRU. pages 554–560.2017

[9] *P. Haffner, G. Tur, and J. H. Wright,* Optimizing svms for complex call classification, in Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, vol. 1. IEEE, 2003, pp. I–632.

[10] *D. Guo, G. Tur, W.-t. Yih, and G. Zweig,* Joint semantic utterance classification and slot filling with recursive neural networks, in Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, 2014, pp. 554–559.

[11] *B. Liu,and I. Lane.* Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling, in arXiv preprint arXiv: 1609.01454 (2016).

[12] *C.-W Goo,G. Gao,Y.-K Hsu,C.-L Huo,and T.-C Chen.* Slot-Gated Modeling for Joint Slot Filling and Intent Prediction, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), volume 2, pages 753–757.

[13] C. *Zhang, Y. Li, N. Du, W. Fan, and P. S. Yu.* Joint slot filling and intent detection via capsule neural networks, in CoRR, abs/1812.09471, 2018.

[14] *T.-C Chi, P.-C Chen, S.-Y Su, and Y.-N Chen.* Speaker role contextual modeling for language understanding and dialogue policy learning, in Proceedings of IJCNLP. pages 163–168 .2017

[15] *S.-Y Su, P.-C Yuan, and Y.-N Chen.* How time matters: Learning time-decay attention for contextual spoken language understanding in dialogues, in NAACL, volume 1, pages 2133– 2142.2018

[16] *J. Kim and J.-H Lee,* Decay-function free time-aware attention to context and speaker indicator for spoken language understanding, in Proceedings of NAACL-HLT, Volume 1 (Long and Short Papers), 2019, pp. 3718–3726.

[17] *K.-S Tai, R. Socher, and C. D. Manning.* Improved semantic representations from tree-structured long short-term memory networks, in Proc. ACL.2015

[18] *J. Pennington, R. Socher, and C. D. Manning.* Glove: Global vectors for word representation. in Proceedings of EMNLP. volume 14, pages 1532–1543.2014

[19] *S. Kim, L. F. DHaro, R. E. Banchs, J. D. Williams, and M. Henderson,* The fourth dialog state tracking challenge, in Proceedings of IWSDS, 2016.

[20] *D. P. Kingma and J. Ba.* Adam: A method for stochastic optimization, in Proceedings of 3rd International Conference on Learning Repre- sentations (ICLR).2018