ATTENTION-BASED DIALOG STATE TRACKING FOR CONVERSATIONAL INTERVIEW COACHING

Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang and Chu-Kwang Chen

Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan

ABSTRACT

This study proposes an approach to dialog state tracking (DST) in a conversational interview coaching system. For the interview coaching task, the semantic slots, used mostly in traditional dialog systems, are difficult to define manually. This study adopts the topic profile of the response from the interviewee as the dialog state representation. In addition, as the response generally consists of several sentences, the summary vector obtained from a long short-term memory neural network (LSTM) is likely to contain noisy information from many irrelevant sentences. This study proposes a sentence attention mechanism combining the sentence attention weights from a convolutional neural tensor network (CNTN) and the topic profile by selectively focusing on significant sentences for attention-based dialog state tracking. This study collected 260 interview dialogs consisting of 3,016 dialog turns for performance evaluation. A five-fold cross validation scheme was employed and the results show that the proposed method outperformed the semantic slot-based baseline method.

Index Terms— Interview coaching system, attention model, LSTM-based autoencoder

1. INTRODUCTION

Over the past few decades, spoken dialogue systems (SDS) have been popular with the people who need some extra help, and have been extensively developed in a variety of areas, such as ticket booking, hotel reservations, interview coaching [1-4], etc. An interview coaching system tries to simulate an interviewer to provide mock interview practice simulation sessions for the users. Most current interview coaching systems focused on different characteristics based on the application purposes. For example, TARDIS [5] aimed to improve the social skills of young people, with a focus on emotional computing. Regarding the coaching system with a fixed scenario, MACH [6] analyzed a user's nonverbal behaviors, such as facial expressions, voice going up or down, head movements, smiling, and eye contact. At the end of a dialog flow, the system provided a summary feedback, indicating which nonverbal behaviors need to be improved. Although all of these coaching systems were used to improve people's interview skill, few of them considered semantic and contextual information of the interview answers, and most of the interview process was pre-defined. It is important that a conversational interview coaching system should take into account the semantic and contextual information of the interviewee's responses for dialog action decisions. Preferably, if the system can understand interviewee's response and ask the questions accordingly, interviewee can practice their interview skills more realistically and effectively. Therefore, this study focuses on how to encode the user's responses into a dialog state and how to track the dialog states based on the interviewee's response.

In a dialog system, DST can help summarize the semantic meaning and the purpose of a user's conversations by finding the corresponding semantic slots for the user to answer. A DST takes as input all of the observable elements up to the current time in a dialog, including the results from the automatic speech recognition and spoken language understanding components [7]. In order to provide a common testbed for the task of Dialog State Tracking, the dialogue state tracking challenge (DSTC) [8] was held by SIGdial, Cambridge University and Microsoft in the past years. However, the interview coaching system is not the same as the traditional task-oriented dialog system, which defines a task as having been completed as long as the specific slots have been completely filled. The questions asked by the interview coaching system depend on the interviewer's decision, and the interviewee's response may affect the next question to be asked. In an interview, the interview coaching system may encounter two problems. The first one is that the interviewee's response is likely to contain noisy information, which is not related to real intent of the response. The second is that the interviewee's response is very diverse and unlikely to label the semantic slots in sentences. In order to solve the above problems, this study proposes a sentence attention-based DST method for the interview coaching system, as shown in Fig. 1. The main contributions of this study are summarized as follows. First, we use the topic model to obtain the topics and the topic-related keywords, which correspond to the semantic slots and slot values, respectively. Second, sentence attention model is proposed to obtain the informative weight of each sentence in interviewee's

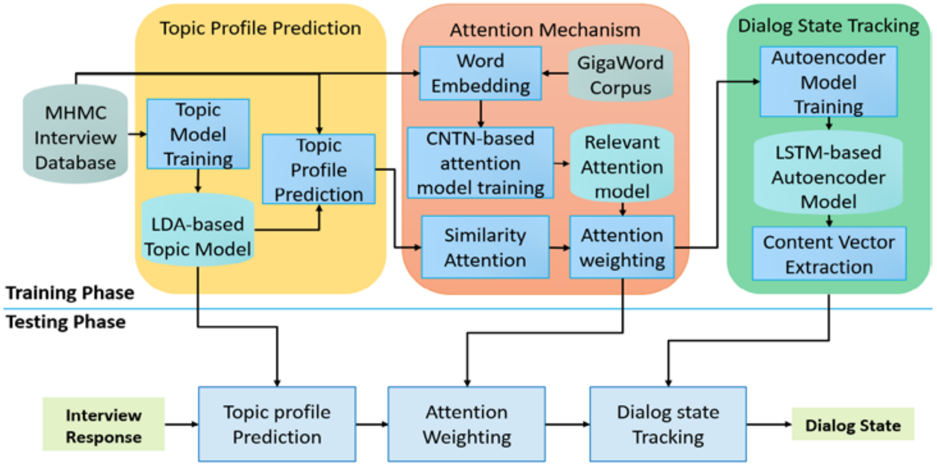

Fig. 1: The block diagram of the proposed interview coaching system.

response. Third, the LSTM-based autoencoder model is employed for dialog state tracking by modeling the sequential relationship between sentences and the temporal transition between dialog turns.

2. INTERVIEW DATABASE COLLECTION

In order to construct an interview coaching system, we invited forty participants to collect the interview database. The question types and topics for the interviews were related to the entrance admission of graduate students. During database collection, every two out of all participants, one serving as the interviewer and the other as the interviewee, had the freedom to complete the interview without using predesigned questions and answers. The interviewee was assigned a random identity background to simulate the real situation. There were two different questions, namely, ordinary questions and follow-up questions. Ordinary questions were not related to the previous question or interviewee's previous response, while follow-up questions were asked based on the interviewee's previous response. Finally, 260 dialogs with 1754 ordinary questions and 1262 follow-up questions were collected to form the NCKU interview database, as shown in Table 1.

Table 1: Details of the NCKU interview database.

	Total
Number of turns	3016
Average number of turns	10.7
Average number of ordinary/follow-up turns	5.74/4.96
Average number of sentences in each answer	3.84
Interview time (minutes) per interview	20

3. SYSTEM FRAMEWORK

3.1 Topic Profile Model

As it is difficult to define all semantic slots in the interview coaching system, this study adopts the topic profile of the response sentences as the representation of a dialog state. Suppose S_m denotes the *m-th* sentence of a dialog turn and there are K topics in the topic model, expressed as $(\lambda_{t1}, \lambda_{t2}, \cdots, \lambda_{tk})$. The topic probability distribution is shown in Eq. (1) used to represent a K-dimensional topic profile vector, in which each element represents the probability of a topic given the *m-th* input sentence.

$$TP_m = \left(P(\lambda_{t1} \mid S_m), P(\lambda_{t2}|S_m), \cdots, P(\lambda_{tk} \mid S_m)\right) \qquad (1)$$

In topic model training phase, word segmentation and stop word filtering for the input sentence are performed first. Then an LDA-based method is adopted to construct the topic detection model [9]. In the testing phase, the interviewee's response sentence is fed to the topic detection model to generate the topic profile for dialog state representation.

3.2 Sentence Attention Mechanism

As the response of the interviewee is composed of several sentences, it may contain some sentences which are considered as redundant or irrelevant. In order to solve this problem, a sentence attention model is proposed to select the informative sentences in the response. This study considers the topic similarity score and the relevance score between a question and its corresponding response to determine the attention weight, as shown in Fig. 2. The topic similarity score is calculated using the cosine similarity, and the relevance score is calculated using the convolutional neural tensor network (CNTN) [10]. The CNTN is composed of a convolutional neural network (CNN) [11] and a neural tensor network (NTN) [12], as shown in Fig. 3. The CNN is used to encode the sentence of the question and the sentence of the response, and the NTN is used to learn the relationship between the question and the response sentences. Given a sentence s, we use GloVe algorithm [13] to obtain the word embedding vector $\mathbf{w}_i \in \mathbb{R}^{n_w}$ for each word w in s. Then we take the word vector $\mathbf{w}_i$ to obtain the input matrix $\mathbf{s} \in \mathcal{R}^{n_w \times l_s}$, where l_s denotes the sentence length. Next, a convolutional layer is obtained by convolving a matrix of weights $\mathbf{m} \in \mathcal{R}^{n \times m}$ with the matrix of activations at the layer below, where m is the filter width. Given a value k and a row vector $\mathbf{p} \in \mathcal{R}^{p}$, we use k-max pooling to select the subsequence $\mathbf{p}_{max}^{p}$ of the k highest values of $\mathbf{p}$. The k-max pooling operation makes it possible to pool the k most active features in $\mathbf{p}$. The final output of CNN is a vector $\mathbf{v}_s \in \mathcal{R}^{n_s}$, which represents the embedding of the input sentence s. Given a question q and its corresponding response r, we can model $\mathbf{v}_q$ and $\mathbf{v}_r$ by using CNN. Then the tensor layer calculates the relevance score of a question-response pair by Eq. (2).

$$s(q,r) = \mathbf{u}^T f\left(\mathbf{v}_q^T \mathbf{M}^{[1:a]} \mathbf{v}_r + \mathbf{V}\begin{bmatrix}\mathbf{v}_q \\ \mathbf{v}_r\end{bmatrix} + \mathbf{b}\right) \qquad (2)$$

where $\mathbf{f}$ is a standard nonlinearity applied element-wise, $\mathbf{V} \in \mathcal{R}^{a \times 2n_s}$, $\mathbf{b} \in \mathcal{R}^{a}$, $\mathbf{u} \in \mathcal{R}^{a}$, $\mathbf{M}^{[1:a]} \in \mathcal{R}^{n_s \times n_s \times a}$ is a tensor and the bilinear tensor product $\mathbf{V}_q^T \mathbf{M}^{[1:a]} \mathbf{v}_r$ results in a vector $h \in \mathcal{R}^{a}$, where each entry is computed by one slice $i = 1, \ldots, a$ of the tensor $h_i = \mathbf{V}_q^T \mathbf{M}^i \mathbf{v}_r$.

Finally, we use the topic similarity score and the relevance score between a question and its response sentences, subject to linear combination and normalization, to obtain the attention weight of the sentence.

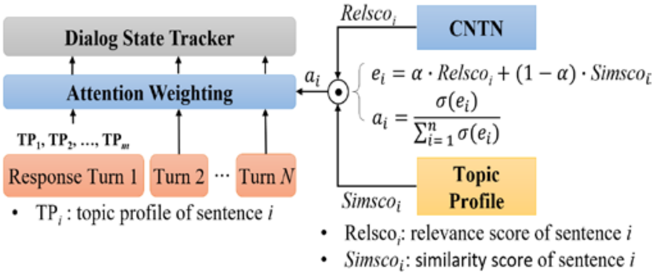
Dialog State Tracker
Attention Weighting
$TP_1, TP_2, \ldots, TP_m$
Response Turn 1
Turn 2
Turn N
a_i
$Relsco_i$
CNTN
$e_i = \alpha \cdot Relsco_i + (1-\alpha) \cdot Simsco_i$
$a_i = \frac{\sigma(e_i)}{\sum_{i=1}^{n} \sigma(e_i)}$
$Simsco_i$
Topic Profile
• TP_i : topic profile of sentence i
• $Relsco_i$: relevance score of sentence i
• $Simsco_i$: similarity score of sentence i

Fig. 2: The framework of sentence attention mechanism.

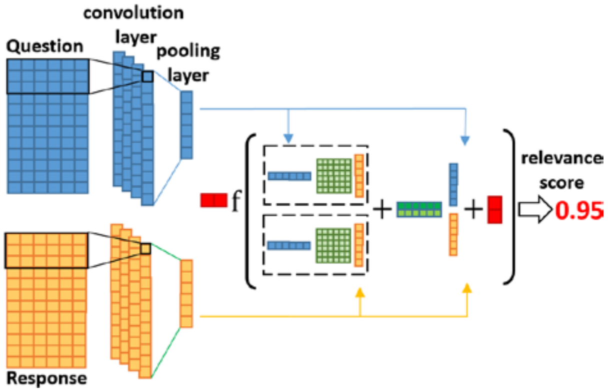

Fig. 3: The CNTN-based relevance score model.

3.3 Dialog State Tracking Model

The dialog state tracker considers the information from the history of the user's response and models the relationship between the current state and dialog history to output the summary state of the interview. In order to obtain the relationship between the sentences in the dialog turns, this study uses the LSTM-based autoencoder [14] to model the relationship between the dialog turns. The LSTM-based autoencoder consists of two LSTMs: an encoder and a decoder, as shown in Fig. 4. The encoder encodes the input sequence into a context vector. The decoder cell initializes the value of the first hidden vector with the context vector. In the autoencoder training process, the mean square error (MSE) between the output sequence and the input sequence is calculated. As the input sequence passes through the encoder, a context vector is obtained as a representation of the input sequence. The decoder uses the context vector to generate the input sequence. In other words, a well-trained autoencoder could encode the input sequence into a context vector which could well represent the input sequence. After the training is completed, the context vector of the encoder is regarded as the representation vector of the input sequence for further process.

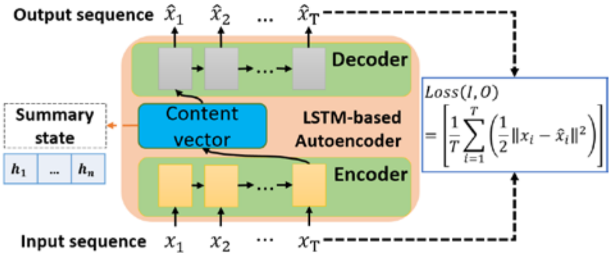
Output sequence $\hat{x}_1$ $\hat{x}_2$ … $\hat{x}_T$
Decoder
Summary state
h_1 … h_n
Content vector
LSTM-based Autoencoder
$Loss(I,O) = \left[\frac{1}{T}\sum_{i=1}^{T}\left(\frac{1}{2}\|x_i - \hat{x}_i\|^2\right)\right]$
Encoder
Input sequence x_1 x_2 … x_T

Fig. 4: The LSTM-based Autoencoder model.

In this study, the dialog state tracker is composed of a two-layer LSTM-based autoencoder, as shown in Fig. 5; the first layer is used to establish the sequential relationship between the sentences in a dialog turn, and the second layer is used to model the temporal relationship between dialog turns.

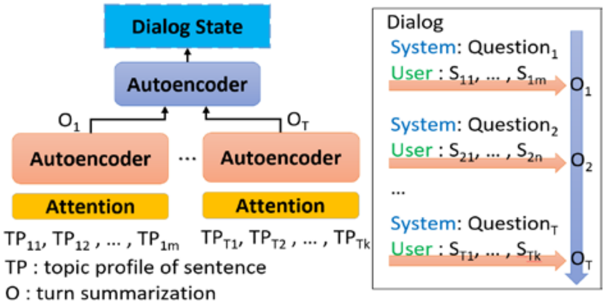
Dialog State
Autoencoder
O_1
O_T
Autoencoder
Autoencoder
Attention
Attention
$TP_{11}, TP_{12}, \ldots, TP_{1m}$
$TP_{T1}, TP_{T2}, \ldots, TP_{Tk}$
TP : topic profile of sentence
O : turn summarization
Dialog
System: $Question_1$
User : $S_{11}, \ldots, S_{1m}$
O_1
System: $Question_2$
User : $S_{21}, \ldots, S_{2n}$
O_2
…
System: $Question_T$
User : $S_{T1}, \ldots, S_{Tk}$
O_T

Fig. 5: The framework of the dialog state tracker

4. EXPERIMENTAL RESULTS

In this study, five-fold cross validation was used to evaluate the performance of the CNTN-based attention model and the proposed method for DST was compared with the traditional semantic slot methods.

4.1 Performance of CNTN-based Attention Model

When the input sequence is composed of several sentences, it may contain some irrelevant sentences. This study used CNTN to calculate the relevance scores among question-response pair sentences. To train the CNTN-based attention model, the numbers of collected relevant and irrelevant question-response pairs were 8,669 and 23,157, respectively. When the tensor dimensionality was 1 and the number of CNN filters was 128, the best correct rate achieved 89.87%, as shown in Fig. 6.

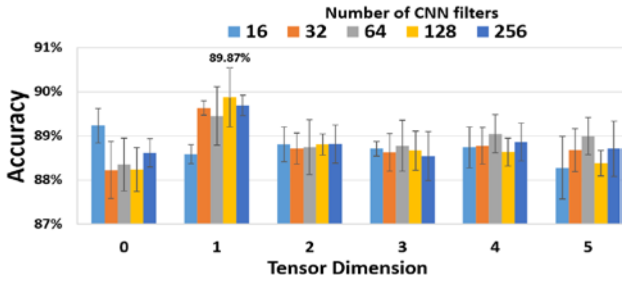

Fig. 6: Effect of CNN filter number and NTN tensor dimension.

4.2 Evaluation on the LSTM-based Autoencoder

In this experiment, we evaluated the LSTM-based autoencoder by visualizing the vector representation of the input and output sequences [15]. The result was obtained when the number of topics was 30. As shown in Fig. 7, the horizontal axis of this input sequence is the dimension of the topic probability distribution vector with sentence attention. The vertical axis is the number of sentences in this dialog, in which the gray level represents the probability of a topic. A brighter gray level implies higher probability. It can be seen from the figure that when the input dialog turn contained more sentences including redundant sentences (12 sentences in this example), the visualization of the representation vector of the output sequence was contaminated by some noises, such as (A) → (A′). Contrast to (A), when the sentences were attentively selected (10 sentences), the visualization of the vector of the output sequence is very similar to that of the input sequence, such as (B) → (B′).

4.3 Evaluation of System Performance

To compare the proposed method with the traditional method based on semantic slots, 10 semantic slots and 10 dialog actions were manually selected to implement the semantic slot-based method. First, we used EHowNet to calculate the similarity of the preceding word and the succeeding word, and then used Affinity Propagation algorithm [16] for word clustering. By manually adjusting the parameters, we finally selected 10 categories as the semantic slots. We also developed a dialog poly decision model with double Q-learning algorithm [17]. The context vector concatenating the historical information was used to predict the next dialog action through deep reinforcement learning. Then we evaluated the performances of the traditional method and the proposed method. The results are shown in Table 2, which lists only the best results for all parameter combinations. In Table 2, it can be seen that the results from the semantic slot-based method were very similar to the method using the topic profile without sentence attention. However, the traditional method requires manual definition of semantic slots and the corresponding slot values, while the proposed method based on the LDA-based topic model could automatically find the topic probability distribution for sentence representation. As the results are very similar, the proposed method still has the advantage. This study argues that in dialog state tracking, the input with an entire answer turn may contain irrelevant sentences. It is beneficial to improve the DST performance by attentively select the informative sentences. The experimental results showed that sentence representation with sentence attention could achieve a better result.

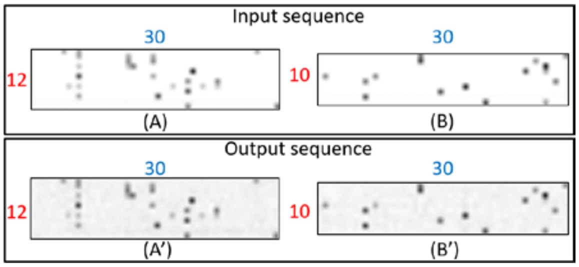

Fig. 7: Visualization result of input sequence and output sequence.

Table 2: Performance Comparison between the traditional and the proposed methods

Method	Semantic slot (baseline)	Topic profile w/o sentence attention	Topic profile w/ sentence attention
Parameters	10 semantic slots Dialog state = 256	30 topic profiles Dialog state = 64	α = 0.5 10 topic profiles Dialog state = 128
Turn	4.74	4.80	5.72
Diff	1.00	0.94	0.02
Accumulative Reward	6.46	6.34	**7.27**

5. CONCLUSIONS

The main purpose of this study is to develop a dialog state tracker for an interview coaching system. This study adopts the topic profiles of the response sentences for dialog state representation and a sentence attention mechanism is proposed by combining a CNTN and a topic detection model for sentence attention weight estimation. Finally, an LSTM-based autoencoder is adopted to model the transition and accumulation of dialog states by selectively focusing on significant sentences for attention-based dialog state tracking. Performance was evaluated using a self-collected interview database based on a five-fold cross validation scheme. The results showed that the proposed method achieved a better performance compared to the semantic slot-based baseline method.

REFERENCES

[1] Y. N. Chen, A. Celikyilmaz, and D. Hakkani-Tür, "Deep Learning for Dialogue Systems," in the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Conada, 2017.

[2] M. H. Su, K. Y. Huang, T. H. Yang, K. J. Lai, and C. H. Wu, "Dialog State Tracking and action selection using deep learning mechanism for interview coaching," in 2016 International Conference on Asian Language Processing (IALP), Tainan, Taiwan, 2016.

[3] M. H. Su, C. H. Wu, K. Y. Huang, T. H. Yang, and T. C. Huang, "Dialog State Tracking for Interview Coaching Using Two-Level LSTM," in 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), Tianjin, China, 2016.

[4] C. H. Wu, M. H. Su, and W. B. Liang, "Miscommunication handling in spoken dialog systems based on error-aware dialog state detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 9, 2017.

[5] H. Jones and N. Sabouret, "TARDIS-A simulation platform with an affective virtual recruiter for job interviews," in IDGEI (Intelligent Digital Games for Empowerment and Inclusion), Chania, Crete, Greece, 2013.

[6] M. E. Hoque, M. Courgeon, J. C. Martin, B. Mutlu, and R. W. Picard, "Mach: My automated conversation coach," in the 2013 ACM international joint conference on Pervasive and ubiquitous computing, Zurich, Switzerland, 2013.

[7] J. Williams, A. Raux, and M. Henderson, "The dialog state tracking challenge series: A review," *Dialogue & Discourse*, vol. 7, no. 3, pp. 4-33, 2016.

[8] J. Williams, A. Raux, D. Ramachandran, and A. Black, "The dialog state tracking challenge," in the SIGDIAL 2013 Conference, Metz, France, 2013.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.

[10] X. Qiu, and X. Huang, "Convolutional Neural Tensor Network Architecture for Community-Based Question Answering," in IJCAI, Buenos Aires, Argentina, 2015.

[11] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1-20, 2016.

[12] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in Advances in Neural Information Processing Systems, Nevada, USA, 2013.

[13] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in EMNLP, Doha, Qatar, 2014.

[14] V. Wan, Y. Agiomyrgiannakis, H. Silen, and J. Vit, "Google's Next-Generation Real-Time Unit-Selection Synthesizer using Sequence-To-Sequence LSTM-based Autoencoders," in Interspeech, Stockholm, Sweden, 2017.

[15] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in nlp," arXiv preprint arXiv:1506.01066, 2015.

[16] X. Li, L. Zhang, L. Wang, and X. Wan, "Effects of BOW Model with Affinity Propagation and Spatial Pyramid Matching on Polarimetric SAR Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 7, pp. 3314-3322, 2017.

[17] H. V. Hasselt, "Double Q-learning," in Advances in Neural Information Processing Systems, Quebec, Canada, 2014.