Leveraging Speaker Profile and Knowledge Enrichment for End-to-End Advising Response Ranking

Sung-Ping Chang Yun-Nung Chen Ting-Rui Chiang Chao-Wei Huang

Abstract

Retrieval-based dialogue modeling has drawn a lot of attention due to its practical usage. Also, different speakers may have diverse preferences and play different roles, so modeling speaker behaviors is useful to better improve dialogue models. This paper focuses on modeling speaker profile information and enriching the entity-related knowledge in dialogues to better predict the next response given the dialogue contexts. The experiments show that the proposed BERT-based ranking model achieves better performance than the provided baseline and adding the speaker and knowledge information further improves the response ranking performance, demonstrating the effectiveness of the proposed framework. The speaker profile modeling and knowledge enrichment approaches are flexible for diverse models, and the future direction is to investigate the capability of generalization to different model architectures.

Introduction

Dialogue-style interaction, where the machine responds to a human based on the conversational contexts, has become an important research topic in recent years. A common method of dialogue modeling is to produce responses using a two-stage selection-based framework. In the first stage, a list of candidates are retrieved by keyword-matching or similarity-based methods. Prior research showed that the list has a high probability to contain the correct response (Ganhotra, Patel, and Fadnis 2019; Chen and Wang 2019). The second stage then aims to select the most appropriate response among the retrieved candidates. Track 1 (Chulaka Gunasekara and Lasecki 2019) of DSTC7 (Yoshino et al. 2018) explores applying the two-stage framework for a two-agent conversation. Following the success of DSTC7, Track 2 in DSTC8 explores applying the framework in conversations involving more than two participants (Seokhwan Kim 2019).

Two datasets are provided in DSTC7. The Ubuntu dataset contains dialogues related to solving technical problems in the Ubuntu system. The advising dataset contains dialogues related to advising course selection. In DSTC7, model performance on the advising dataset is generally worse than the performance on the Ubuntu dataset. The result indicates that the advising dataset is more challenging than the Ubuntu dataset. Therefore, we focus on the advising dataset in this work.

In recent years, many attempts have been made to select responses with deep neural networks. Before the invention of contextualized word embeddings such as ELMo (Peters et al. 2018) and BERT (Devlin et al. 2019), most deep neural models include two encoders and a scorer. One encoder is to encode a given partial conversation, and the other encoder is to encode each candidate respectively. Afterward, the scorer scores each candidate based on the encoded vectors. Some prior work utilizes two RNNs as the encoders (Feng et al. 2015; Mueller and Thyagarajan 2016; Lowe et al. 2015). Later work further improves the encoding process using attention mechanisms (Wan et al. 2015; Bahdanau, Cho, and Bengio 2014; Tan et al. 2015; Rocktäschel et al. 2016; Wang, Liu, and Zhao 2016; Santos et al. 2016; Shen, Yang, and Deng 2017; Tay, Tuan, and Hui 2018; Chen and Wang 2019). Among them, the ESIM model (Chen and Wang 2019) achieves the best performance in DSTC7. Besides these models, Vig and Ramea shows promising results using BERT-based models.

In spite of the success of previous models, only a few of them used data other than the dialogs in the advising dataset, such as personal preferences or suggested courses. Some work leveraged the student profile by matching the tokens in the dialogue with the student's suggested courses as additional features (Vig and Ramea 2019; Huang et al. 2019; Chiang et al. 2019). Ganhotra, Patel, and Fadnis encoded each suggested course into a sentence representation and incorporated it into the dialogue representation with an attention layer. Sun et al. encoded the additional information with RNNs and treated them as long-term memories in their model.

In our work, we explore ways to incorporate the extra unstructured data given in the advising dataset with BERT. We proposed three different methods to leverage the provided information:

- Replace the course number in the conversation with the course name to capture the semantic information in the course name.

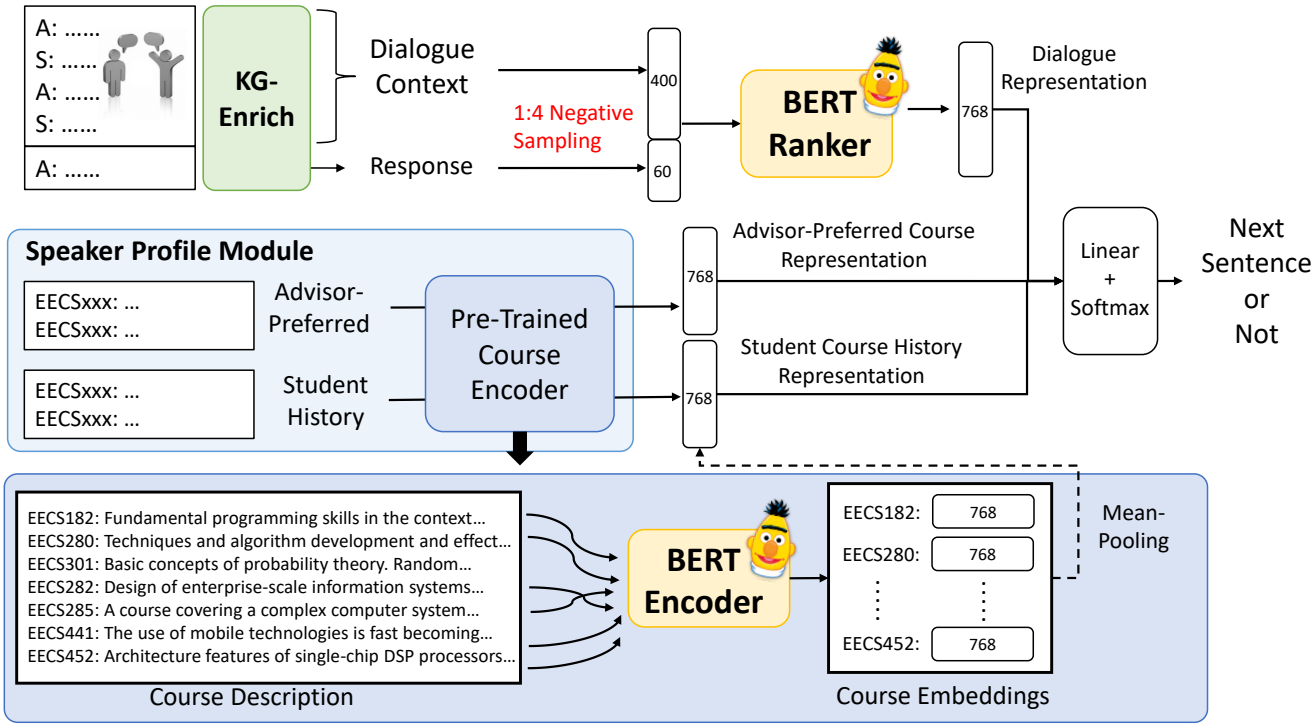

Figure 1: The illustration of the proposed framework.

- Fuse the embedding of the description of taken courses with dialogue representation to leverage the information of taken courses.
- Fuse the embedding of the description of suggested courses with dialogue representation to leverage the information of suggested courses.

On top of them, we also conducted comprehensive experiments to identify the BERT layers best suited for response selection. We show them in the experiments section.

Problem Formulation

In the response selection challenge, given a partial conversation and a set of response candidates, the goal is to rank the responses based on the probability of being the next sentence. In addition to ranking, the system is expected to detect whether the correct response does not appear in the candidate pool.

We formulate the response selection task as a sequence-pair classification problem. We denote a partial conversation consisting of l utterances as U: $\{u_1, u_2, ..., u_l\}$, where each utterance u_i is a sequence of words $\{w_{i,1}, w_{i,2}, \cdots, w_{i,|u_i|}\}$. We prepend a special token, either $\langle$advisor$\rangle$ or $\langle$student$\rangle$, to the utterances spoken by the associated speaker in order to indicate the speaker identity for each utterance. Therefore, the speaker information can be modeled to further improve the response ranking model. A candidate set consisting of k response candidates is defined as X: $\{x_1, x_2, ..., x_k\}$; note that among all candidates, there may be zero or one correct response. The labels indicating whether each corresponding candidate is the correct answer are denoted as Y: $\{y_1, y_2, ..., y_k\}$, where

$$y_i = \begin{cases} 1, & x_i \text{ is the correct answer} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In the classification model, given a partial conversation and a response candidate, the objective is to estimate the correct label (next response or not).

Proposed Response Ranking Framework

In order to rank the responses based on the given dialogue contexts, we built a response ranker based on *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al. 2019). The BERT model is a pre-trained language model that uses the Transformer architecture (Vaswani et al. 2017) and is capable of performing sequence-pair classification after fine-tuning on labeled sequence pairs. Here we borrow BERT's powerful capability of modeling the semantics of sequences to estimate whether a response is the next sentence given the dialogue contexts. The proposed approach is illustrated in Figure 1, where the basic architecture is a response ranker, a knowledge enrichment model and a speaker profile module are the additional mechanism our paper proposes, which are highlighted in the figure. Below we describe each component in detail.

For each candidate x_j, we concatenate all utterances in dialogue contexts as the first sequence, $d = [u_1, u_2, ..., u_l]$, and use the response candidate, x_j, as the second sequence. The [SEP] token is added at the end of both sequences to rep-

resent utterance boundaries. We concatenate these two sequences, d and x_j, and add the [CLS] token at the beginning as the input to our model, denoted as s_j. The model takes s_j as the input and outputs a representation of the whole sequence h_j with dimension d_{h_j} that is the hidden state of the [CLS] token. This setting is similar to the two-sentence classification task such as NLI in the original BERT paper (Devlin et al. 2019). After fine-tuning the BERT model, the probability of x_j being the correct answer is calculated as:

$$e_j = \sum W_h h_j, \quad (2)$$

$$p(x_j) = \texttt{softmax}(e_j), \quad (3)$$

where $W_h \in \mathbb{R}^{d_{h_j} \times 2}$ are trainable parameters, $p(x_j)_1$ is the probability that x_j is the next sentence, and $p(x_j)_2$ is the probability that x_j is a random sentence.

In addition, our model needs to handle the situation where there is no correct response. To consider this situation, our model calculates the entropy of the next response's probability in the candidate set. Entropy is used to express the uncertainty of the message; if the entropy is higher, the uncertainty of the information is higher accordingly. High entropy suggests the higher probability of no accurate response in the candidates. The entropy and the probability of not having a correct response is calculated as:

$$E_n = \sum_{j=1}^{k} p(x_j) \log p(x_j), \quad (4)$$

$$P_n = \frac{1}{\alpha} \times \frac{E_n - \min(E_n)}{\max(E_n) - \min(E_n)}, \quad (5)$$

where α is a trainable parameter, $\min(E_n)$ equals to 0, and $\max(E_n)$ depends on the number of candidates.

The binary cross entropy function is adopted as the loss function:

$$\mathcal{L}(U, X, Y) = \sum_{j=1}^{k} \big(y_j \log p(x_j) + (1 - y_j) \log(1 - p(x_j))\big). \quad (6)$$

Then the model is expected to estimate whether the response candidate is the correct next sentence given the partial conversation and whether there is no accurate response in the whole candidate set.

Knowledge Enrichment for Entities

The advising dataset provides rich metadata about the entities (the course names and their descriptions), which may provide informative cues for better modeling the dialogues. Hence, our model focuses on utilizing the metadata to add more rich details about the courses mentioned in the conversations. To do so, we detect the course numbers mentioned in the conversations and then add their course names to the utterances to explicitly enrich the course-related information.

We conduct several sets of experiments using heuristics such as replacing all class IDs with class titles and adding a class title after a class ID. A real-world challenge we found was that when a class ID appears many times in the conversation, the sequence length may exceed the input limit of the BERT model if we add the class title to every class ID frequently and some important information about the original utterance may be missing. To avoid this, we design an alternative approach that adds the class title after the class ID only when the ID appears for the first time in the conversation. This idea is inspired by how people talk in real life; when someone introduces a new term to others for the first time, they often describe it in detail using its full name. After that, they may only use a short name or even an abbreviation for convenience. In our model, the connecting signal can be modeled because we bridge the full name and the abbreviation.

Moreover, since a class ID sometimes appears as a standalone number in the utterance, the model may link the number information with the associated course ID due to the lack of the department information. To explicitly allow the model to link such information, we add the department information such as "EECS" before the number if this number appears in a list of the EECS department classes.[1] An example of knowledge enrichment is illustrated in Figure 2.

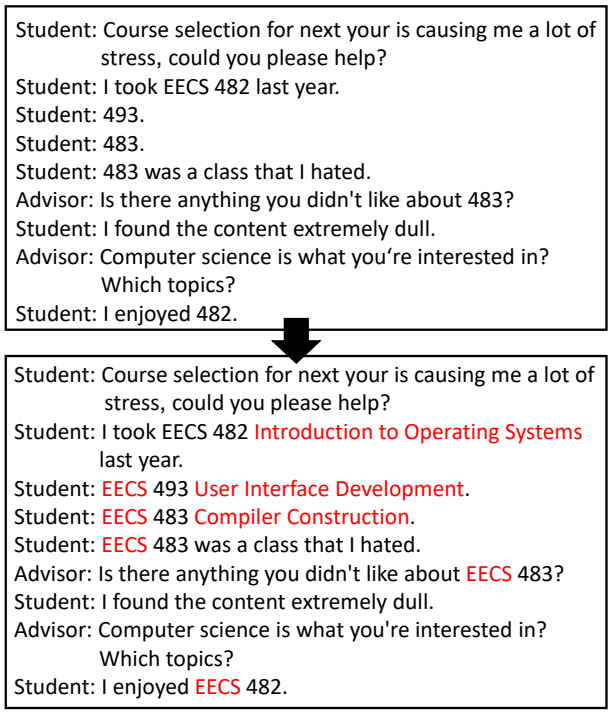

Figure 2: An example of knowledge enrichment for entities. The upper part in the figure is the original conversation utterances, and the bottom part of the figure is the conversation utterances after adding the course title and the department information.

[1] This design is based on the observation that most conversations are about courses from the EECS department.

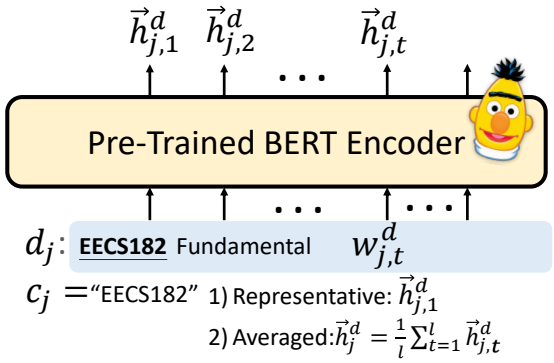

Figure 3: The illustration of how a course embedding is built for two versions.

Speaker Profile Module

In addition to the course names and descriptions, other rich information is provided in the advising dataset. Each conversation also includes the suggested courses from the advisor consisting of k courses ID: $C_{\text{suggested}}$: $\{c_1^{\text{suggested}}, c_2^{\text{suggested}}, ..., c_k^{\text{suggested}}\}$, and the prior courses the student took consisting of m courses ID: C_{prior}: $\{c_1^{\text{prior}}, c_2^{\text{prior}}, ..., c_m^{\text{prior}}\}$. We hypothesize that if we add this data into the model, the model is able to model the students' priority or the advisors' suggestions. In order to explicitly leverage the extra metadata about courses in our model, each course description $d_j = \text{desc}(c_j)$ is encoded by the the pre-trained BERT encoder to form the course embeddings. Here we attempt two ways to form the course-specific embeddings: 1) *representative* – the embeddings of a *single representative token* from the description sequence and 2) *averaged* – the averaged embeddings of *all tokens* from the description sequence. The course embeddings are illustrated in Figure 3. In our preliminary experiments, the average version works better, so we describe the details based on the average version. In the BERT model, we utilize the last layer for building the course embeddings:

$$\vec{h}_{j,t}^{d} = \text{BERT}^{12}(w_{j,t}^{d}), \tag{7}$$

while c_j is the course ID, and mean pooling is applied over the outputs to form the course embeddings based on the description d_j:

$$\vec{r}_j^{d} = \frac{1}{l}\sum_{t=1}^{l}(\vec{h}_{j,t}^{d}), \tag{8}$$

where l is the sequence length of the course description.

When training, because an advisor's suggestions and a student's prior courses can contain multiple courses, mean pooling is applied over the $r_j^{d^{\text{prior}}}$ and $r_j^{d^{\text{prior}}}$ to get the entire representation of the $\text{desc}(C_{\text{suggested}})$ and $\text{desc}(C_{\text{prior}})$:

$$\bar{r}^{d^{\text{prior}}} = \frac{1}{m}\sum(\bar{r}_j^{d^{\text{prior}}}), \tag{9}$$

$$\bar{r}^{d^{\text{suggest}}} = \frac{1}{k}\sum(\bar{r}_j^{d^{\text{suggest}}}). \tag{10}$$

This formulation is illustrated in the lower part of Figure 1. Given the embeddings of speaker-related course profiles, advisor-prederred course representations and student course history representations, we sum them to the last hidden layer of the BERT output, denoted as h_j above, respectively and simultaneously fusing the information of the course description with dialogue representations. The whole model can be trained in an end-to-end manner by 1:4 negative sampling to perform the final prediction about whether this response is the next sentence given the partial conversation as illustrated in Figure 1.

Experiments

To evaluate the performance of the proposed framework, we conduct a set of experiments using the benchmark challenge dataset. The detailed experiments are shown and discussed as follows.

Setup

Here are some of the settings and hyperparameters we used to train the model: We set the maximum length of the conversation to 400, and the maximum length of the candidate response is set to 60. We used a batch size of 8 and fine-tune for 3 to 4 epochs over the data, and tried the learning rate from 1e-5 to 3e-5. The optimizer we used is Adam, and the proportion of negative sampling during training is 1:4.

Baseline Systems

Dual Encoder (Lowe et al. 2015): uses two LSTMs with tied weights to encode the context $d = u_1, u_2, ..., u_l$ and the response x into fixed-length representations c, r, respectively. The final hidden state of the LSTM is used to represent an input word sequence. The probability of x being the next utterance of c is then calculated as

$$p = \sigma(c^T M r + b)$$

where the matrix M and bias b are learned parameters.

Results

To illustrate the improvement and utility of our proposed approach and features, we compare the performance between our model and the baseline systems. Table 1 shows the empirical results on the development set of the subtask 1.

Effectiveness of Knowledge Enrichment As shown in the Table 1, we found that the overall performance is better when we just set up the entity enrichment and did not add any course description at all. However, if we add the settings about course description and entity enrichment at the same time, the result gets worse in most conditions. The reason we supposed is that the additional title adding for entity enrichment may cause the length exceed the sequence length limit of BERT even if we just add a title for each class ID once in the conversation. Thus, the utterance will lose some important information about the original utterance and the information maybe have a strong relationship with the course description.

	Layers	SC	PC	EE	R@1	R@2	R@5	R@10	MRR
Baseline					22.18	33.60	49.31	62.20	35.51
Proposed					24.0	35.4	53.0	67.2	37.87
				✓	**25.0**	**36.8**	54.6	67.4	38.81
	1	✓			24.0	36.6	54.6	67.8	38.47
		✓		✓	22.0	34.6	56.6	67.8	36.85
			✓		23.4	35.4	55.6	67.6	37.86
			✓	✓	22.8	35.6	54.8	67.2	37.35
		✓	✓		23.4	35.0	55.0	68.6	37.82
		✓	✓	✓	23.2	36.4	55.2	65.4	37.56
	4	✓			24.6	36.4	**57.6**	69.0	**38.91**
		✓		✓	23.6	34.2	55.8	65.4	37.52
			✓		23.8	33.4	54.6	67.4	37.42
			✓	✓	22.6	33.0	54.2	68.6	36.89
		✓	✓		24.0	35.2	56.0	69.0	38.27
		✓	✓	✓	23.4	36.4	56.8	**70.6**	38.42

Table 1: Results of our experiments on subtask 1 development sets (%). SC stands for description of suggested courses, PC stands for description of prior courses, and EE stands for entity enrichment.

Last Utterance	Student: Is Python taught in EECS183?
True Answer	Advisor: Python is required in many courses, it is also widely used today.
BERT	Advisor: Python is required in many courses, it is also widely used today.
+ Advisor Suggested	Advisor: Python is required in many courses, it is also widely used today.
+ Student Prior	Advisor: Python is required in many courses, it is also widely used today.
+ Both	Advisor: Python is required in many courses, it is also widely used today.
Last Utterance	Student: Could I do EECS482 next term?
True Answer	Advisor: I see here you have to take EECS281 first.
BERT	Advisor: I recommend EECS492, It has an average workload and teaches you about Artificial Intelligence.
+ Advisor Suggested	Advisor: There's a large workload in EECS482, but you'll be taught a great deal about operating systems.
+ Student Prior	Advisor: There's a large workload in EECS482, but you'll be taught a great deal about operating systems.
+ Both	Advisor: There's a large workload in EECS482, but you'll be taught a great deal about operating systems.
Last Utterance	Student: The classes I'm taking next semester is what I'm figuring out right now.
True Answer	Advisor: Do you have any specific interests or classes you wish to take mind?
BERT	Advisor: This semester, if you're taking 281, then next semester you can being taking upper level classes.
+ Advisor Suggested	Advisor: Is there a specific type of course are you looking for?
+ Student Prior	Advisor: Next semester what are you looking to do?
+ Both	Advisor: If you wish to graduate in 4 years, it is important that you take EECS 281 even though it is difficult, as it is an important prerequisite class.

Table 2: Qualitative analysis for three types of cases.

Effectiveness of Speaker Profile As shown in the Table 1, we find that the model fused with the suggested course descriptions gives the better performance and the model fused with prior course descriptions does not show any improvement but even slightly worse. The probable reason is that all predictions are from the advisor instead of the student, so that modeling the advisor's profile is much important than modeling the student's. To further investigate how our model performs, we divide all conversations into three categories for deep discussions. The first one, also the simplest, is when the last utterance in the conversation is a simple question, and the model can select the correct response easily, because it does not need to consider the previous conversation. The second one is the question which needs to use some information may be mentioned previously to select response correctly. The third one, as the most difficult one, is when the last utterance is such a declarative sentence. Unless the model has comprehensive understanding about the previous utterance, the profile of teachers and students, and see lots of cases during training. Otherwise, it is difficult for even

	Layers	SC	PC	EE	R@1	R@2	R@5	R@10	MRR
Fixed	1	V			24.0	36.6	54.6	67.8	38.47
Fine-Tune	1	V			25.0	36.6	54.0	67.2	38.82

Table 3: Results of our experiments for fine-tune the course description in our BERT model.

	R@1	R@2	R@5	R@10	MRR
Submitted	19.2	–	34.2	43.4	27.1
Revised	23.2	34.0	52.2	64.8	36.3

Table 4: Testing results for our proposed method.

people to select the correct response from the candidate set. Sometimes, it does not mean that the answer select from the model is incorrect although it is not the same as the correct response. Table 2 shows examples of three categories and how our speaker profile module provides additional cues for selecting suitable candidates. It is clear that the predicted responses in the third case are also reasonable; this scenario tells the challenges of the current evaluation and the misalignment between the evaluation score and the actual performance in terms of human perspective.

Representations in Different Layers For contextual embeddings, we learned from previous work where the sum of the last four hidden layers in BERT would have a better performance than only the last hidden layer in BERT on some tasks (Devlin et al. 2019). Therefore, we want to verify whether different layers of the course description would have an impact on this task when other settings are fixed. As shown in Figure 4, we can see that it will have an improvement both on Recall 10 and MRR from last layer to sum the last four layers when only sum the suggest course description, it declines a little bit both on Recall 10 and MRR when only sum the prior course description, and it also has improved on Recall 10 and MRR when summing both course description. Obviously, when we fuse the sum of the last four layers of course description, each setting has more change based on the improvement or declination from baseline to sum the last layer of course description. Thus, we can say that the sum of the last four hidden layers in BERT indeed has better representation than only the last hidden layer in BERT in the advising task.

Fine-Tuned BERT on Course Descriptions We also tried adding the descriptions of suggested and prior courses into our BERT Ranker to fine-tune the model. We hypothesize that there are specific relationships between the course descriptions and the utterance that the pre-trained BERT Encoder has never seen. We select the settings that sum the last hidden layer of suggested course description with BERT Ranker output (row 4 of Table 1) to verify our speculation and observe their change of the performance. The results show in Table 3. We found that the impact on the experiment is not significant, its score improved on Recall 1 and MRR but declined on Recall 5 and Recall 10. We thought that because we need to use more GPU memory to fine-tune our model, we set the maximum length of the course description

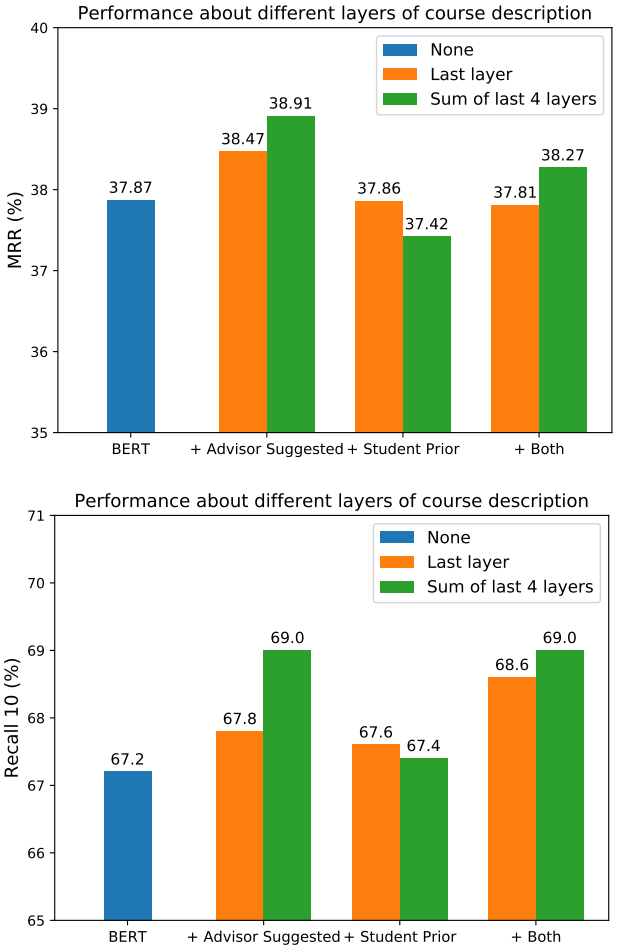

Figure 4: Performance of course description representation in different layers.

to 200 but some of the course descriptions are longer than the maximum length. This may cause the description is not representative enough and also leads to no improvement in scores.

Testing Performance In the DSTC8 challenge, the proposed systems are submitted for official evaluation (Seokhwan Kim 2019). However, due to the misunderstanding of the uploading rules about returning "NONE", the official scores are much worse than the actual performance. Therefore, we revise the submitted version and re-evaluate the approach by the published test set. The evaluation results are shown in Table 4, and the scores are around the 5-th place among 10 participants.

Conclusions

This paper proposes an approach that leverages the speaker profile information for better modeling the response selection task. Specifically, in the advising conversations between students and advisors, advisors' suggestions and students' prior courses should be considered and may benefit the decision of which response is better given the current dialogue history. Our method models the course descriptions for capturing their semantics and enriches the course-related knowledge in order to improve the response ranker. The comprehensive experiments demonstrate the effectiveness of the proposed approach. We thought that there are still some experiments that can be conducted and discussed on this task in the future such that the performance of individual layers, how the ratio of negative sampling impacts the model, what impact does the cut-off have when the length of conversation exceed our maximum length etc. Considering that the concept about leveraging speaker profiles is flexible, the future work plans to investigate whether the proposed method can generalize to diverse tasks about dialogue modeling.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chen, Q., and Wang, W. 2019. Sequential matching model for end-to-end multi-turn response selection. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7350–7354. IEEE.

Chiang, T.-R.; Huang, C.-W.; Su, S.-Y.; and Chen, Y.-N. 2019. Learning multi-level information for dialogue response selection by highway recurrent transformer. *arXiv preprint arXiv:1903.08953*.

Chulaka Gunasekara, Jonathan K. Kummerfeld, L. P., and Lasecki, W. S. 2019. Dstc7 task 1: Noetic end-to-end response selection. In *7th Edition of the Dialog System Technology Challenges at AAAI 2019*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; and Zhou, B. 2015. Applying deep learning to answer selection: A study and an open task. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, 813–820. IEEE.

Ganhotra, J.; Patel, S. S.; and Fadnis, K. 2019. Knowledge-incorporating esim models for response selection in retrieval-based dialog systems. In *The 7th Dialog System Technology Challenge (DSTC7)*.

Huang, C.-W.; Chiang, T.-R.; Su, S.-Y.; and Chen, Y.-N. 2019. Rap-net: Recurrent attention pooling networks for dialogue response selection. *arXiv preprint arXiv:1903.08905*.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Mueller, J., and Thyagarajan, A. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, 2227–2237.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kocisky, T.; and Blunsom, P. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*.

Santos, C. d.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Seokhwan Kim, Michel Galley, C. G. S. L. A. A. B. P. H. S. J. G. J. L. M. A. M. H. L. L. J. K. K. W. S. L. C. H. A. C. T. K. M. A. R. X. Z. S. S. R. G. 2019. The eighth dialog system technology challenge. *arXiv preprint*.

Shen, G.; Yang, Y.; and Deng, Z.-H. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1179–1189.

Sun, S.; Tam, Y.-C.; Cao, J.; Yan, C.; Fu, Z.; Niu, C.; and Zhou, J. 2019. End-to-end gated self-attentive memory network for dialog response selection. In *The 7th Dialog System Technology Challenge (DSTC7)*.

Tan, M.; Santos, C. d.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Multi-cast attention networks for retrieval-based question answering and response prediction. *arXiv preprint arXiv:1806.00778*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Vig, J., and Ramea, K. 2019. Comparison of transfer-learning approaches for response selection in multi-turn conversations. In *The 7th Dialog System Technology Challenge (DSTC7)*.

Wan, S.; Lan, Y.; Guo, J.; Xu, J.; Pang, L.; and Cheng, X. 2015. A deep architecture for semantic matching with multiple positional sentence representations. *arXiv preprint arXiv:1511.08277*.

Wang, B.; Liu, K.; and Zhao, J. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1288–1297.

Yoshino, K.; Hori, C.; Perez, J.; D'Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B.; Gao, S.;

Marks, T. K.; Parikh, D.; and Batra, D. 2018. The 7th dialog system technology challenge. *arXiv preprint*.