Computer Speech & Language 63 (2020) 101079

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

RAP-Net: Recurrent Attention Pooling Networks for Dialogue Response Selection

Chao-Wei Huang[1], Ting-Rui Chiang, Shang-Yu Su, Yun-Nung Chen Ph.D.*

National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan

ARTICLE INFO

Article History:
Received 26 July 2019
Revised 10 January 2020
Accepted 21 January 2020
Available online 29 February 2020

Keywords:
Attention
Dialogue modeling
Response selection
DSTC

ABSTRACT

The response selection has been an emerging research topic due to the growing interest in dialogue modeling, where the goal of the task is to select an appropriate response for continuing dialogues. To further push the end-to-end dialogue model toward real-world scenarios, the seventh Dialog System Technology Challenge (DSTC7) proposed a challenge track based on real chatlog datasets. The competition focuses on dialogue modeling with several advanced characteristics: (1) natural language diversity, (2) capability of precisely selecting a proper response from a large set of candidates or the scenario without any correct answer, and (3) knowledge grounding. This paper introduces *recurrent attention pooling networks* (RAP-Net), a novel framework for response selection, which can well estimate the relevance between the dialogue contexts and the candidates. The proposed RAP-Net is shown to be effective and can be generalize across different datasets and settings in the DSTC7 experiments.

1. Introduction

With the increasing trend about dialogue modeling, response selection and generation have been widely studied in the NLP community. In order to further evaluate the current capability of machine learning models, a benchmark dataset was proposed in the seventh Dialog System Technology Challenge (DSTC7) (D'Haro et al., 2020), where the task is to select the most probable response given a partial conversation. To approximate the real world scenarios, several variants of selections are investigated in this task: (1) selecting from 100 candidates, (2) selecting from 120,000 candidates, (3) selecting multiple answers, (4) there may be no answer, and (5) with external information. In addition, the ability of generalization should be examined; hence, two datasets, Ubuntu IRC dialogs (Kummerfeld et al., 2018) and course advising corpus, are utilized for the experiments. These datasets have very different properties, where the dialogs in the Ubuntu IRC dialogs dataset are very technical, and are more problem-solving-oriented, while the dialogs in the course advising dataset tend to be more casual, and the goals are more like inquiring information rather than solving a specific problem. The course advising dataset also comes with metadata about participants' profiles and course information, which can be leveraged as domain-specific knowledge to enhance dialogue modeling. In sum, the challenge covers a wide range of scenarios in real-world applications and serves as a set of benchmark experiments for evaluating dialogue response selection models.

Recently, deep neural networks have been widely adopted for end-to-end response selection modeling. The prior work generally employed two encoders to map the conversation and the response into vector representations, and then designed a classifier to measure the relation between these two representations. An intuitive design is to encode two sequences separately via recurrent neural networks (RNNs) and then compute a score between the last hidden state of two RNNs (Feng et al., 2015; Mueller and Thyagarajan,

*Corresponding author.
E-mail address: y.v.chen@ieee.org (Y.-N. Chen).
[1] First two authors have equal contribution.

https://doi.org/10.1016/j.csl.2020.101079
0885-2308/

2016; Lowe et al., 2015). The MV-LSTM (Wan et al., 2015) improved the design by deriving a similarity matrix between outputs of RNNs, and then used max-pooling and multi-layer perceptron (MLPs) to aggregate the similarity scores. To better utilize the interactive information, other approaches employed the attention mechanism (Bahdanau et al., 2014) to facilitate the encoding process (Rocktäschel et al., 2016; Shen et al., 2017; Tan et al., 2015; Wang et al., 2016; Santos, Tan, Xiang, Zhou; Tay et al., 2018).

Motivated by the prior work that effectively utilized attention mechanisms in diverse ways, this paper proposes a novel framework for dialogue response selection, called *recurrent attention pooling networks* (RAP-Net). The proposed model consists of (1) a feature extractor which uses the architecture of multi-cast attention network (MCAN) (Tay et al., 2018) for extracting features from input words, (2) feature-fusion layer for integrating domain-specific knowledge-grounded features and information from the MCAN layer, and (3) a proposed dynamic pooling recurrent layer for extracting sentence-level information by pooling dynamically based on utterance boundaries. The proposed model is shown to be effective for different datasets (Ubuntu, Advising) and different settings (subtask 1, 3, 4) in the DSTC7 experiments.

2. Task description

In the response selection challenge, given a partial conversation and a set of response candidates, the system is required to select one response from the set of candidates. A partial conversation consists of l consecutive utterances $U : \{u_1, u_2, \ldots, u_l\}$, where each utterance is a sequence of words $u_i : \{w^U_{i,0}, w^U_{i,1}, w^U_{i,2}, \ldots, w^U_{i,n_i}\}$. We use special identifiers <speaker1> and <speaker2> to represent the speaker identities, and the special identifier of the corresponding speaker is prepended to the beginning of the utterances. A candidate set consists of k candidates, denoted as $X : \{x_1, x_2, \ldots, x_k\}$, where each candidate is a sequence of words $x_j : \{w^X_{j,1}, w^X_{j,2}, \ldots, w^X_{j,m_j}\}$. For the course advising dataset, each dialogue comes with profiles of the participants, and we use this information to extract knowledge-grounded features for each input word. The knowledge-grounded features of a word w are denote as $F(w)$. The extraction of $F(w)$ will be detailed in Section 4.3. Among the candidates, there will be either some correct responses or none. The labels indicating if the candidate are correct answers are denoted as $Y : \{y_1, y_2, \ldots, y_k\}, y_i \in \{0, 1\}$.

3. RAP-Net: recurrent attention pooling networks

In this paper, we propose a novel framework, recurrent attention pooling networks (RAP-Net), for dialogue response selection, as illustrated in Fig. 1. The four-step pipeline is described as follows:

3.1. Multi-Cast Attention Network (MCAN)

The multi-cast attention network (MCAN) (Tay et al., 2018) aimed at extracting extra word features with various attention mechanisms. We apply MCAN as an additional feature extractor. Here we concatenate context utterances $d = [u_1, u_2, \ldots, u_l]$ as the first sequence, while $q = x_j$ as the sequence for the jth candidate. For all words in either d or q, the MCAN passes the word representation into a highway layer H (Srivastava et al., 2015) to obtain a new representation w':

$$w' = H(w) = \sigma(W_g w) \odot \mathrm{ReLU}(W_h w) + \left(1 - \sigma(W_g w)\right) \odot w, \tag{1}$$

where W_g, W_h are parameters to learn, $\odot$ denotes element-wise multiplication between vectors and σ is the sigmoid function.

Then MCAN applies several attention mechanisms on two word sequences to capture the interaction between words, the attention mechanisms include:

- *Intra-attention* aims to capture interaction of words in the same sequence. For a sequence d, a similarity matrix S is calculated as

$$s_{i,j} = w'^T_i M w'_j, \tag{2}$$

  where w'_i and w'_j are representations of the ith and jth word of d respectively, and M is a learned transform matrix. For w'_j, the attention are then used to compute a weighted sum over contexts to form a new representation:

$$w'_{intra} = \sum_i \frac{\exp(s_{i,j})}{\sum_k \exp(s_{i,k})} w'_i, \tag{3}$$

  where the weights are computed by performing softmax operation over columns of the similarity matrix S, and the intra-attention results for q are calculated similarly.
- *Inter-attention* aims to capture interaction between two word sequences. A similarity matrix S is calculated as

$$s_{i,j} = w'^T_{d,i} M w'_{q,j}, \tag{4}$$

  where $w'_{d,i}$ and $w'_{q,j}$ are representations of the ith and jth word of d and q respectively. Three pooling mechanisms are utilized to obtain different features from inter-attention results, as illustrated in Figs. 2 and 3 M is not shared across different pooling mechanisms. The pooling functions include:

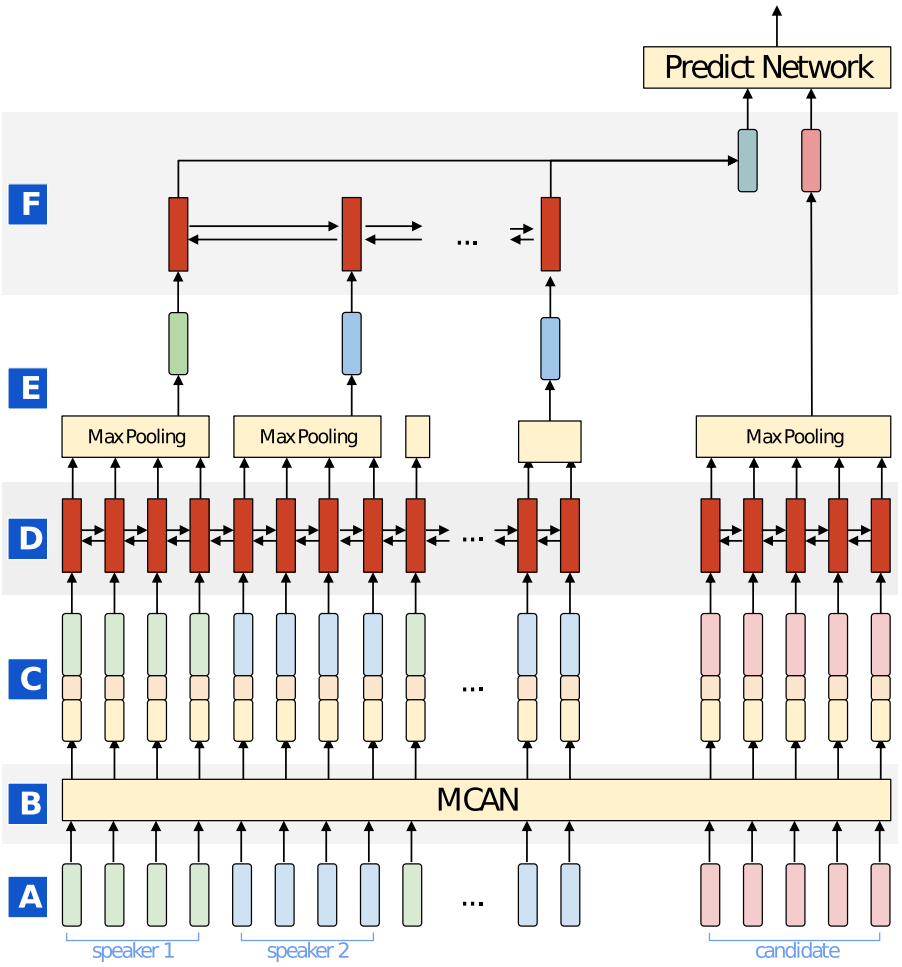

Fig. 1. The architecture of the whole proposed model. (A) is the word embedding of utterances (light green and light blue color denotes words spoken by different speakers) and candidate (light red color). (B) is the MCAN. (C) Word embeddings along with extra features, which include knowledge grounded features (light orange color) and features extracted by the MCAN (light yellow color). (D) is the first bi-directional LSTM layer. (E) is the dynamically pooling layer. Note that LSTM outputs are grouped according to utterances. (F) is the second bi-directional LSTM layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *max-pooling*: the attention results are calculated as

$$w'_{max} = \begin{cases} \text{softmax}\left(\max_{col}(S)\right)^T q & \text{if } w \in q \\ \text{softmax}\left(\max_{row}(S)\right) d & \text{if } w \in d \end{cases} \tag{5}$$

- *mean-pooling*: the attention results are calculated as

$$w'_{mean} = \begin{cases} \text{softmax}\left(\text{mean}_{col}(S)\right)^T q & \text{if } w \in q \\ \text{softmax}\left(\text{mean}_{row}(S)\right) d & \text{if } w \in d \end{cases} \tag{6}$$

- *alignment-pooling*: the attention results are calculated as

$$w'_{align} = \begin{cases} \sum_i \frac{\exp(s_{i,j})}{\sum_k \exp(s_{i,k})} w'_{d,i} & \text{if } w = w_{q,j} \\ \sum_j \frac{\exp(s_{i,j})}{\sum_k \exp(s_{k,j})} w'_{q,j} & \text{if } w = w_{d,i} \end{cases} \tag{7}$$

Finally, we can have a feature vector of twelve scalar features by interacting w' with 4 attention results using 3 different compression method: concatenation, subtraction and element-wise multiplication:

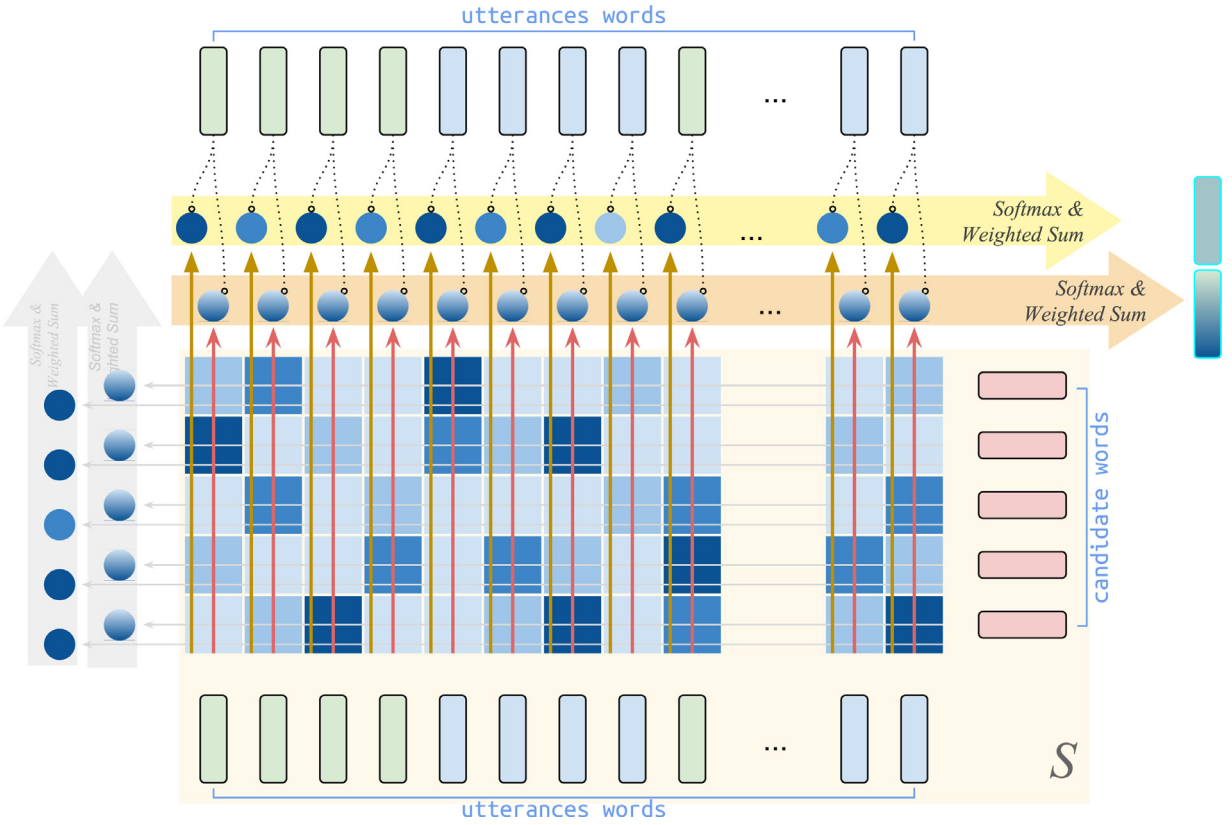

Fig. 2. Illustrator of inter-attention with mean and max pooling. The brown and violet color arrows are column-wise max and mean operation respectively. The output vectors are the summation of utterances word embeddings weighted by the mean and max values. The dotted lines at the top of the figure denote that the word embeddings (green and blue rounded rectangles) are weighted by the values (blue circles). Weighted summation of the candidate word vectors (pink round corner rectangles) is omitted for simplicity here. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$
\begin{aligned}
f_{mcan}(w') = [\ & W_1[w'; w'_{align}]; && W_2[w'_{align} - w']; && W_3[w'_{align} \odot w']; && W_4[w'; w'_{intra}]; \\
& W_5[w'_{intra} - w']; && W_6[w'_{intra} \odot w']; && W_7[w'; w'_{mean}]; && W_8[w'_{mean} - w']; \\
& W_9[w'_{mean} \odot w']; && W_{10}[w'; w'_{max}]; && W_{11}[w'_{max} - w']; && W_{12}[w'_{max} \odot w'];\ ],
\end{aligned} \tag{8}
$$

where W_i are learned compression matrices that map a vector into a scalar, and $[\,\cdot\,;\,\cdot\,]$ denotes vector concatenation.

3.2. Word feature augmentation

In the second stage, each word w is augmented by concatenating the domain specific knowledge-grounded features $F(w)$ and features extracted by MCAN $f_{mcan}(w)$ after the word embeddings. The augmented representation of w is denoted as

$$\tilde{w} = [w; F(w); f_{mcan}(w)]. \tag{9}$$

3.3. Dynamic pooling recurrent networks

We propose dynamic pooling recurrent network which contains two layers of recurrent units and one dynamic pooling layer between the two recurrent layers to encode contextual information. In our model, a first-level bi-directional LSTM $LSTM^1$ is employed (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997), which focuses on encoding the utterance-level information as hierarchical recurrent neural networks (HRED) does (Serban et al., 2016). The ith sequence ($i = 1, 2, \ldots, l$) is encoded as

$$\overrightarrow{h}^1_{i,t}, \overrightarrow{c}^1_{i,t} = \overrightarrow{\text{LSTM}}^1(\tilde{w}^U_{i,t-1}, \overrightarrow{h}^1_{i,t-1}, \overrightarrow{c}^1_{i,t}), \tag{10}$$

$$\overleftarrow{h}^1_{i,t}, \overleftarrow{c}^1_{i,t} = \overleftarrow{\text{LSTM}}^1(\tilde{w}^U_{i,t+1}, \overleftarrow{h}^1_{i,t+1}, \overleftarrow{c}^1_{i,t}), \tag{11}$$

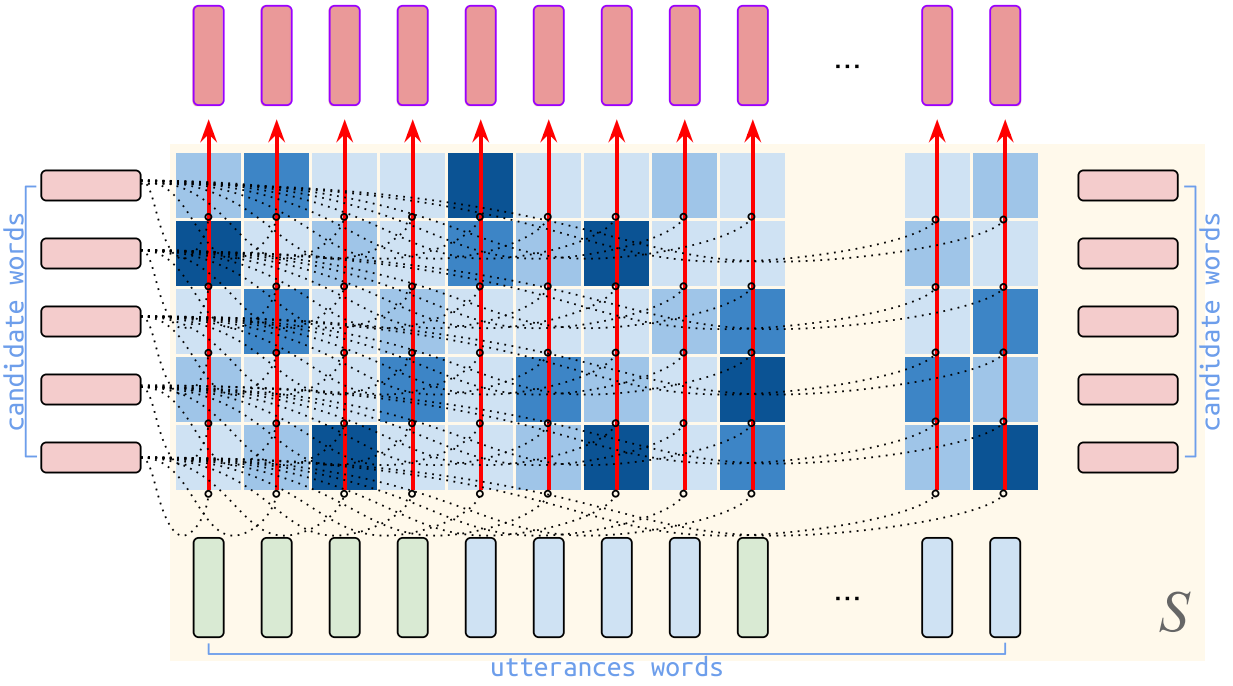

Fig. 3. Illustration of inter-attention with alignment pooling. The red arrows implies summation of the candidates (pink round corner rectangles) weighted by the value of the column. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where h and c are the hidden states and cell states respectively. Different from HRED, which encodes each utterance separately, the first layer of our dynamic pooling recurrent network encodes utterances by concatenating the utterances as a single sequence. Therefore, the initial recurrent hidden state of an utterance is the last hidden state from the previously encoded utterance:

$$\overrightarrow{h}^1_{i,0}, \overrightarrow{c}^1_{i,t} = \overrightarrow{h}^1_{i-1,n_i}, \overrightarrow{c}^1_{i-1,n_i} \tag{12}$$

$$\overleftarrow{h}^1_{i,0}, \overleftarrow{c}^1_{i,t} = \overleftarrow{h}^1_{i+1,n_i}, \overleftarrow{c}^1_{i+1,n_i} \tag{13}$$

In the second part of the dynamic pooling recurrent network, the dynamic pooling layer is used to generate one vector representation $\hat{h}^1_i$ for each utterance u_i by pooling dynamically based on the utterance length n_i over the encoded hidden states from the first bidirectional recurrent layer:

$$\hat{h}^1_i = [\max([\overrightarrow{h}_{i,1}; \overleftarrow{h}_{i,1}], \ldots, [\overrightarrow{h}_{i,n_i}; \overleftarrow{h}_{i,n_i}]); \text{mean}([\overrightarrow{h}_{i,1}; \overleftarrow{h}_{i,1}], \ldots, [\overrightarrow{h}_{i,n_i}; \overleftarrow{h}_{i,n_i}])], \tag{14}$$

where max(.) and mean(.) denote the operations of max pooling and mean pooling of the vectors over dimensions. Finally, there is another bi-directional LSTM layer, $LSTM^2$, which encodes utterance-level representations:

$$\overrightarrow{h}^2_i, \overrightarrow{c}^2_i = \overrightarrow{\text{LSTM}}^2(\hat{h}^1_i, h^2_{i-1}, c^2_{i-1}), \tag{15}$$

$$\overleftarrow{h}^2_i, \overleftarrow{c}^2_i = \overleftarrow{\text{LSTM}}^2(\hat{h}^1_i, h^2_{i+1}, c^2_{i+1}), \tag{16}$$

and the last LSTM cell state is used as the dialogue-level representation:

$$r^C = [\overrightarrow{c}^2_l; \overleftarrow{c}^2_1]. \tag{17}$$

3.4. Candidate selection

Each candidate x_j is encoded by the first LSTM layer in the dynamic pooling recurrent network, $LSTM^1$:

$$\overrightarrow{h}^x_{j,t}, \overrightarrow{c}^x_{j,t} = \overrightarrow{\text{LSTM}}^1(\tilde{w}^U_{j,t-1}, \overrightarrow{h}^x_{j,t-1}, \overrightarrow{c}^x_{j,t}), \tag{18}$$

$$\overleftarrow{h}^x_{j,t}, \overleftarrow{c}^x_{j,t} = \overleftarrow{\text{LSTM}}^1(\tilde{w}^U_{j,t+1}, \overleftarrow{h}^x_{j,t+1}, \overleftarrow{c}^x_{j,t}), \tag{19}$$

and both max pooling and mean pooling are applied over the outputs to get the candidate representation:

$$r^X_j = [\max([\overrightarrow{c}^x_{j,1}; \overleftarrow{c}^x_{j,1}], \ldots, [\overrightarrow{c}^x_{j,m_j}; \overleftarrow{c}^x_{j,m_j}]); \text{mean}([\overrightarrow{c}^x_{j,1}; \overleftarrow{c}^x_{j,1}], \ldots, [\overrightarrow{c}^x_{j,m_j}; \overleftarrow{c}^x_{j,m_j}])]. \tag{20}$$

Then the probability of the candidate x_j being the correct response is calculated as

$$p(x_j) = \sigma\Big(H_1\Big(H_2([r^C; r^X_j; r^C \odot r^X_j; r^C - r^X_j])\Big)\Big), \tag{21}$$

where H_1, H_2 are highway layers (Srivastava et al., 2015) with ReLU activation. The binary cross entropy function is utilized as the objective:

$$L(U, X, Y) = \sum_{j=1}^{k} y_j \log p(x_j) + (1 - y_j) \log\Big(1 - p(x_j)\Big). \tag{22}$$

4. Experiments

To evaluate the performance of the proposed RAP-Net, we conduct experiments on the two datasets provided by DSTC7-Track1, and compare our results with two baseline systems.

4.1. Dataset

DSTC7-Track1 contains two goal-oriented dialogue datasets — (1) Ubuntu data and (2) Advising data. There are five subtasks in this challenge, where this paper focuses on the subtask 1, 3 and 4, because the same model architecture can be applied to these subtasks. Here we briefly describe the settings for each subtask:

- Subtask 1: There are 100 response candidates for each dialogue, and only one is correct.
- Subtask 3: There are 100 response candidates for each dialogue. The number of correct responses ranges from 1 to 5. Multiple correct responses are generated using paraphrases.
- Subtask 4: There are 100 response candidates for each dialogue, and an additional candidate *no answer* should also be considered. The number of correct responses is either 0 or 1.

Note that although we discard subtask 5 which comes with external knowledge, i.e., Ubuntu manual pages, it is possible to incorporate the provided information by learning domain-specific embeddings from it, which can be used as input to our model.

4.2. Baseline systems

- Dual Encoder (Lowe et al., 2015): uses two LSTMs with tied weights to encode the context $d = \{u_1, u_2, \ldots, u_l\}$ and the response x into fixed-length representations c, r, respectively. The final hidden state of 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.
- HRED (Serban et al., 2016): has a similar structure as the Dual Encoder, but uses two LSTMs to encode context hierarchically. Each utterance in the dialogue context is encoded separately by an utterance-level LSTM $LSTM^1$. The encoded representations are then fed into a conversation-level LSTM $LSTM^2$ to produce the context representation c. A response x is encoded by $LSTM^1$ into response representation r. The prediction is calculated similarly as the Dual Encoder above.

4.3. Experimental details

We use pre-trained 300-dimensional word embeddings via fasttext (Mikolov et al., 2018) to initialize the embedding matrix and fix it during training. The word embeddings of out-of-vocabulary (OOV) are initialized randomly. In the advising dataset, the suggested courses $C_{suggested}$ and the prior courses C_{prior} of the student are given along with a conversation. Therefore, to explicitly utilize this knowledge in our model, we extract two features for each word and then concatenate them as the knowledge-grounded features, $F(w)$:

$$F_1(w) = \begin{cases} 1 & \text{if } w \in C_{suggested} \\ 0 & \text{otherwise.} \end{cases}$$

$$F_2(w) = \begin{cases} 1 & \text{if } w \in C_{prior} \\ 0 & \text{otherwise.} \end{cases}$$

$$F(w) = [F_1(w); F_2(w)]$$

Note that there is no additional knowledge provided for the Ubuntu dataset except for the subtask 5. Therefore, only $f_{mcan}(w)$ is added to the word representations.

We use adam as our optimizer to minimize the training loss (Kingma and Ba, 2014). We utilize LSTM as the recurrent cell, and the hidden layer size of LSTM is set to 128. The initial learning rate varies from 0.001 to 0.0001, which is a hyperparameter for tuning. We train our models for 10 epochs and select the best-performing model based on the development set.

Following the official evaluation metrics, we use *recall at 10* (R@10) and *mean reciprocal rank* (MRR) to report the performance of our models. The final score is the average of two metrics.

4.4. Results

To explicitly validate the effectiveness of the proposed model and auxiliary 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.

Dynamic pooling recurrent networks Rows (b) and (c) show that our dynamic pooling recurrent networks (DP-LSTM) outperforms HRED in terms of all metrics on both datasets, especially on the advising dataset. The results show that concatenating utterances into a single sequence can benefit conversation encoding.

MCAN feature Adding $f_{mcan}(w)$ as an auxiliary feature (row (d)) further improves the performance by a large margin on the Ubuntu dataset, yielding a 23.5% relative improvement. It demonstrates that MCAN feature also helps our model achieve better results on the advising dataset.

Knowledge-grounded feature For advising dataset, we extract a 2-dimensional knowledge-grounded feature $F(w)$ to enhance word representations. As shown in Table 1, adding $F(w)$ (row (e)) yields a 56.4% relative improvement, which is significantly greater than the improvement of adding $f_{mcan}(w)$. The results show the difficulty of solving this task on the advising dataset without any prior knowledge. The effectiveness of our knowledge-grounded feature $F(w)$ shows that identifying course names is crucial for this dataset. The best model on the advising dataset is the DP-LSTM with both MCAN and knowledge-grounded features added to the input word representations (row (f)), achieving about 66% and 60% average performance for Ubuntu and advising datasets respectively.

4.5. Official evaluation

In the DSTC7 challenge, the proposed systems are submitted for official evalution. For each subtask, our submitted system consists of several models with different hyperparameters and auxiliary features. Using different features gives our model multiple perspectives to the data and hence improves the prediction accuracy. The official evaluation results are shown in Table 2. In the official evaluation, the superior performance and the achieved rankings across different subtasks clearly demonstrate the effectiveness of the proposed model. Considering that our rankings are either 2 or 3 among 20 teams, we argue that the proposed RAP-Net can successfully estimate the relatedness between dialogues and responses and generalize across different datasets.

4.6. Ablation study

To further understand the contribution of each component, we conduct an ablation test on the RAP-Net model. Table 3 shows the ablation results on the Ubuntu subtask 1 development set. We remove one component in a test and evaluate the resulting model using R@10 and MRR. Note that after removing dynamic pooling, the last hidden state of an LSTM is used as the sequence-level representation, so this setting is equivalent to HRED with an additional feature f_{mcan}. We also ablate mean pooling and max pooling from Eqs. (14) and (20) to verify their effectiveness.

Table 1
Results on subtask 1 development sets (%).

		Ubuntu			Advising		
		R@10	MRR	Average	R@10	MRR	Average
Baseline	(a) Dual Encoder	62.5	36.23	49.39	25.8	11.81	18.81
	(b) HRED	65.2	37.87	51.56	39.2	18.68	28.94
RAP-Net	(c) DP-LSTM	66.3	41.26	53.81	49.0	21.99	35.49
	(d) DP-LSTM+f_{mcan}	**76.7**	**56.18**	**66.45**	51.0	25.80	38.40
	(e) DP-LSTM+$F(w)$	–	–	–	72.9	38.07	55.50
	(f) DP-LSTM+f_{mcan}+$F(w)$	–	–	–	**76.6**	**42.84**	**59.72**

Table 2
The official testing results of our submitted systems. Two different test sets for advising dataset are provided. Note that in subtask 3, only advising dataset is provided for training and evaluating. The rankings of this challenge are based on the average score.

Task	Ubuntu				Advising Case 1			Advising Case 2			
	R@10	MRR	Avg	Rank	R@10	MRR	Avg	R@10	MRR	Avg	Rank
Subtask 1	81.0	64.86	72.93	3	80.4	49.14	64.77	61.0	30.61	45.81	2
Subtask 3	–	–	–	–	68.4	39.34	53.87	60.4	31.71	46.05	3
Subtask 4	84.1	63.17	73.63	2	84.2	45.31	64.75	64.0	30.70	47.35	3

Table 3
Ablation results on Ubuntu development set (%).

Model	R@10	MRR	Average
DP-LSTM + f_{mcan}	76.7	56.18	66.45
- inter-attention	69.4	43.92	56.66
- intra-attention	76.3	55.86	66.08
- highway encoder	75.8	54.94	65.37
- dynamic pooling	76.1	55.18	65.67
- mean pooling	75.8	55.02	65.40
- max pooling	76.1	54.99	65.54

The ablation results show that the inter-attention is the most crucial component to our model, because the average score drops drastically by almost 10% if it is removed, demonstrating the importance of modeling the interaction between the conversation and the response for this task. It is found that removing highway encoder, dynamic pooling, mean pooling, or max pooling results in a reduction of roughly 1% in terms of the average score, so they also contribute to the improvement slightly. Furthermore, the intra-attention benefits least to performance, which is similar to the findings in the prior work (Tay et al., 2018).

4.7. Attention analysis

As described in the previous section, the attention f_{mcan} is a key feature in our framework. To deeply investigate this feature, we examine its numerical value to perform qualitative analysis. An example of attention scores for each word in a sequence is shown in Fig. 4. It can be found that the features extracted by mean-pooling, max-pooling and intra-attention are always equal or close to zero

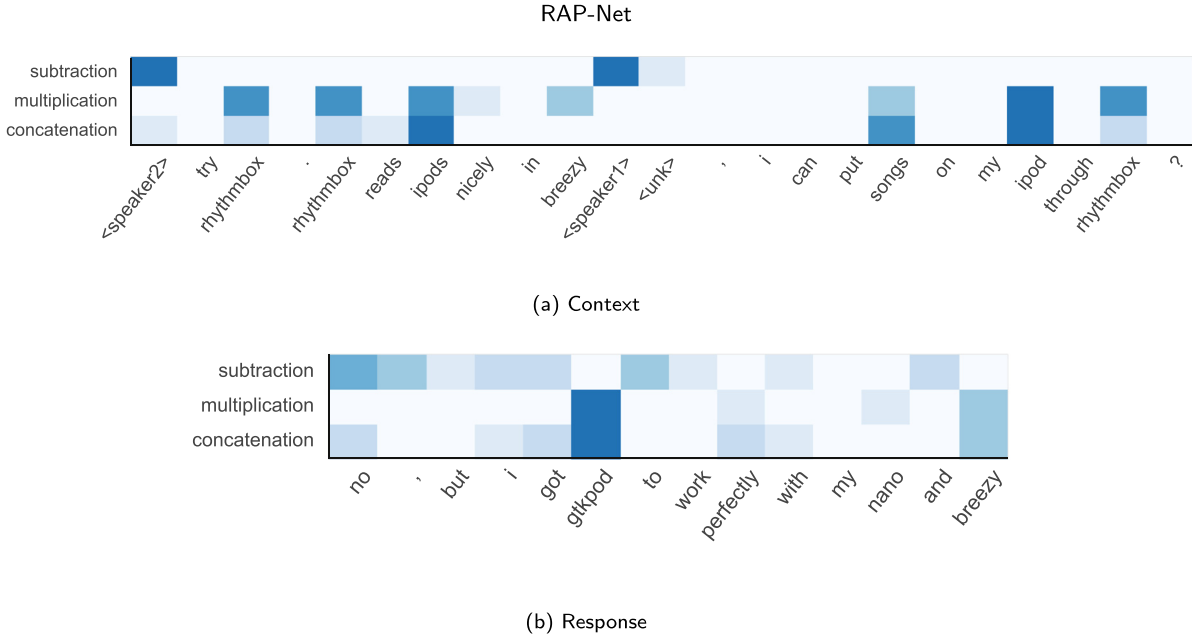

Fig. 4. Visualization of attention scores. We plot the attention scores f_{mcan} of two sequences from Ubuntu development set: (a) A partial conversation and (b) The correct response corresponding to the conversation. The conversation is truncated to the last two sentences due to width limitation. Darker color represents higher attention score. Note that each row is normalized separately since the range of values varies for each dimension.

with no obvious pattern, so we only plot features extracted by alignment-pooling for simplicity. The x-axis indicates the words in the context or response, and the y-axis represents different compression methods described in the MCAN section.

From Fig. 4, we observe that the attention has the ability to model word overlapping between two sequences. For example, the word *breezy* appears in both sequences, and it has a relatively higher attention score. In addition to the ability to model explicit word overlapping, MCAN can also identify words that are relevant to the other sequence. Here MCAN gives *rhythmbox* and *ipod* larger scores than other words in the context, even though they do not appear in the response. The reason is that words such as *gtkpod* and *breezy* in the response are related to *ipod*, so the model correctly identifies the words that are relevant in the context. Similarly, the word *gtkpod* in the response obtains the highest attention score, because it is the most relevant to the context.

The features extracted by multiplication and concatenation shows similar patterns. However, the features extracted by subtraction seems to be only activated by <speaker> and <unk> tokens or other function words. The probable reason is that this dimension assists the encoder to recognize unimportant words. We should note that these observed patterns are not consistent over different runs. Generally there is at least one dimension that models word relevance across sequences, and at least one dimension that recognizes unimportant words.

5. Conclusions

This paper proposes a novel framework, recurrent attention pooling networks (RAP-Net), which focuses on precisely measuring the relations between dialogue contexts and the responses for dialogue response selection. The DSTC7 experiments are conducted to evaluate the proposed model, where multi-cast attention network (MCAN) and our proposed knowledge-grounded features are proved to be useful, and each attention and pooling mechanism is demonstrated to be effective. In sum, RAP-Net is capable of capturing the salient information from dialogues and is good at selecting a proper response for two different types of dialogue data. In the future, the proposed model can be evaluated on other retrieval-based tasks to test the model capability of generalization.

Acknowledgments

This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grants 108-2636-E-002-003 and 108-2634-F-002-019.

References

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

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. In: Proceedings of 3rd International Conference on Learning Representations.

Kummerfeld, J. K., Gouravajhala, S. R., Peper, J., Athreya, V., Gunasekara, C., Ganhotra, J., Patel, S. S., Polymenakos, L., Lasecki, W. S., 2018. A Large-Scale Corpus for Conversation Disentanglement. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3846-3856.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A., 2018. Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Mueller, J., Thyagarajan, A., 2016. Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence.

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

Lowe, R., Pow, N., Serban, I., Pineau, J., 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. Proceedings of 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, p. 285.

Kingma, D. P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.

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

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45 (11), 2673–2681.

Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J., 2016. Building end-to-end dialogue systems using generative hierarchical neural network models.. In: Proceedings of the AAAI, 16, pp. 3776–3784.

Shen, G., Yang, Y., 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, pp. 1179–1189.

Srivastava, R. K., Greff, K., Schmidhuber, J., 2015. Training very deep networks. Proceedings of the Advances in Neural Information Processing Systems 2377–2385.

Tan, M., Santos, C. d., Xiang, B., Zhou, B., 2015. LSTM-based deep learning models for non-factoid answer selection. Proceedings of the International Conference on Learning Representations (ICLR).

Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X., 2015. A deep architecture for semantic matching with multiple positional sentence representations. Proceedings of the AAAI 16, 2835–2841.

Wang, B., Liu, K., 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), 1, pp. 1288–1297.

Tay, Y., Tuan, L. A., Hui, S. C., 2018. Multi-cast attention networks. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2299-2308.

D'Haro, L F, Yoshino, K, Hori, C, Marks, T K, Polymenakos, L, Kummerfeld, J K, Galley, M, Gao, X, 2020. Overview of the seventh Dialog System Technology Challenge: DSTC7. Computer Speech & Language 62. In press.

Chao-Wei Huang is a current M.S. student and holds the bachelor degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan in 2018. His research has been focused on dialogue systems, spoken language understanding, and speech technologies.

Ting-Rui Chiang is a current M.S. student and holds the bachelor degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan in 2018. His research has been focused on dialogue system, question answering, and machine learning.

Shang-Yu Su is a current Ph.D. student in Department of Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan. He holds the bachelor degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan. His research has been focused on deep learning and dialogue systems.

Yun-Nung Chen received Ph.D. in Computer Science from at Carnegie Mellon University, PA. She has been an assistant professor in the Department of Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan. Her research interests include spoken dialogue understanding, speech summarization, information extraction, and machine learning. Dr. Chen received Google Faculty Research Award and NVIDIA Scientific Research Award and currently serves a member in Speech and Language Technical Committee in IEEE.