

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

Chao-Wei Huang* Ting-Rui Chiang* Shang-Yu Su Yun-Nung Chen

National Taiwan University, Taiwan

{r07922069,r07922052,f05921117}@csie.ntu.edu.tw y.v.chen@ieee.org

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.

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 the machine learning models, a benchmark dataset was proposed in the seventh Dialog System Technology Challenge (DSTC7) (Yoshino et al. 2018), 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. Some subtasks are much more difficult than the original setting. 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. 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 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, Cho, and Bengio 2014) to facilitate the encoding process (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).

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) multi-cast attention network (MCAN) (Tay, Tuan, and Hui 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. Furthermore, the framework can also generalize to other retrieval-based tasks.

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 candidates

*Huang and Chiang have equal contributions.
Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

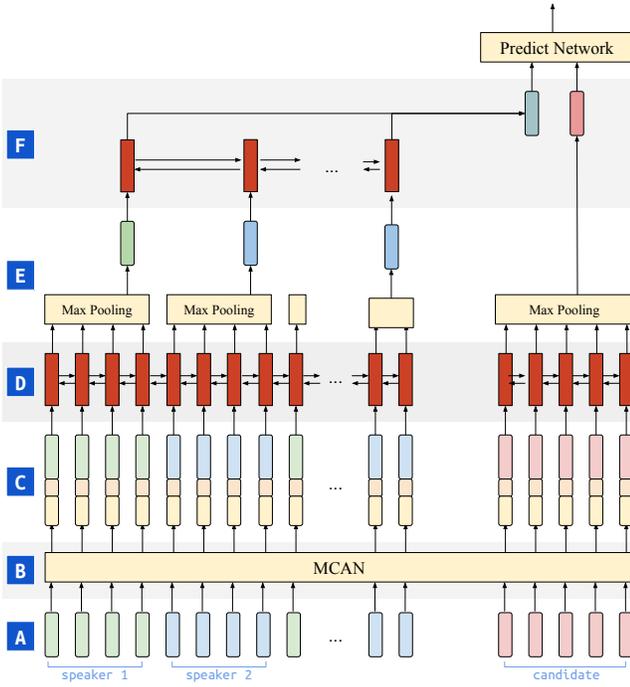


Figure 1: The architecture of the whole proposed model. (A) is the word embedding of utterances (light green and light blue color denotes words spoken by different speakers) and candidate (light red color). (B) is the MCAN. (C) Word embeddings along with extra features, which include knowledge grounded 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.

set. A partial conversation consists of l utterances: $U : \{u_1, u_2, \dots, u_l\}$, an utterance is a sequence of words. Each speaker participated in the conversation is given a special identifier, say $\langle \text{speaker1} \rangle$, $\langle \text{speaker2} \rangle$, and the special identifier is prepended to the utterances which speaker speaks. So the i th utterance is denoted as $u_i : \{w_{i,0}^U, w_{i,1}^U, w_{i,2}^U, \dots, w_{i,n_i}^U\}$. A candidate set consists of k candidates $X : \{x_1, x_2, \dots, x_k\}$. And each candidate is a sequence of words $x_j : \{w_{j,1}^X, w_{j,2}^X, \dots, w_{j,m_j}^X\}$. For some datasets, some word features grounded to specific domain knowledge are also available. The knowledge-grounded features of a word w are denote as $F(w)$. Among the candidates, there would be either some correct responses or none. The labels indicating if the candidate are correct answers are denoted as $Y : \{y_1, y_2, \dots, y_k\}$.

RAP-Net: Recurrent Attention Pooling Networks

In this paper, we propose a novel framework for dialogue response selection illustrated in Figure 1, and the four-step pipeline is described as follows.

Multi-Cast Attention Network (MCAN)

First, the multi-cast attention network (MCAN) (Tay, Tuan, and Hui 2018) is applied to extract extra word features with various attention mechanisms on two word sequences. Here we concatenate context utterances $d = [u_1, u_2, \dots, u_l]$ as the first sequence, while $q = x_j$ as the sequence for the j th candidate. For each word w in either d or q , we pass the word representation into a highway layer H (Srivastava, Greff, and Schmidhuber 2015) to obtain a new representation w' :

$$w' = H(w) = \sigma(W_g w) \odot \text{ReLU}(W_h w) + (1 - \sigma(W_g w)) \odot w.$$

where W_g, W_h are parameters to learn.

Before applying any attention operation, every word will by be transformed with the highway layer, the attention mechanisms include:

Intra-attention for a sequence d , a similarity matrix S is calculated as

$$s_{i,j} = w_i'^T M w'_j,$$

where w_i and w_j are i -th and j -th word of d respectively, and M is a parameter to learn. For w_j , the attention are then used to weighted sum over contexts to form a new representation:

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

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 a similarity matrix S is calculated as

$$s_{i,j} = w'_{d,i}{}^T M w'_{q,j},$$

where $w'_{d,i}$ and $w'_{q,j}$ are i -th and j -th word of d and q respectively. The attention results are calculated differently for different pooling mechanisms (M is not shared across different pooling mechanisms), as illustrated in Figure2 and Figure3:

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

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

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

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

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

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

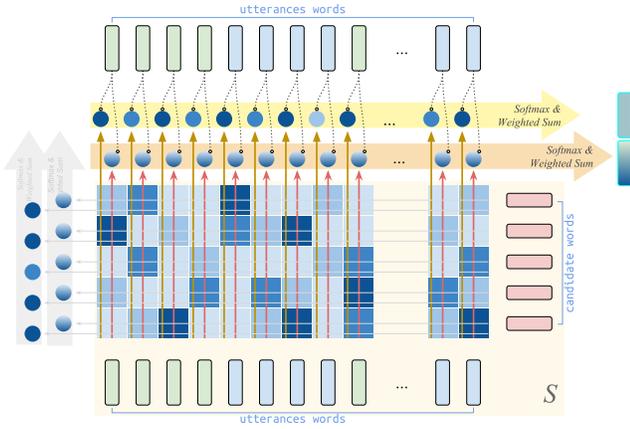


Figure 2: Illustration of inter-attention with mean and max pooling. The brown and violet color arrows are column-wise max and mean operation respectively. The output vectors are the summation of utterances word embeddings weighted by the mean and max values. The dotted lines at the top of the figure 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.

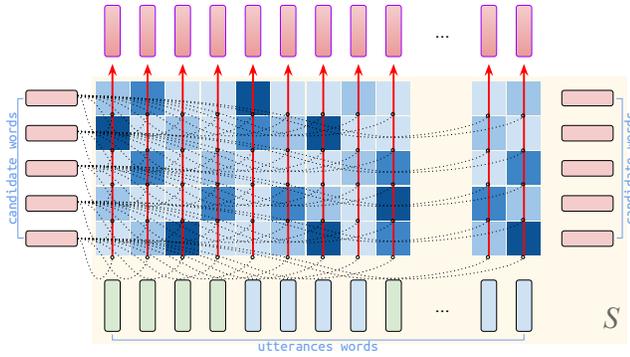


Figure 3: Illustration of inter-attention with alignment pooling. The red arrows implies summation of the candidates (pink round corner rectangles) weighted by the value of the column.

Finally, we can have a feature vector of twelve scalar features by interacting w' with other features produced by attention approaches described above:

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

where W_i are compression matrices that map a vector into a scalar and are parameters to learn.

Word Feature Augmentation

Secondly, each word w is augmented by concatenating the domain specific knowledge-grounded features and features extracted by MCAN after the word embeddings. The sequences of words with extra features in the dialogue contexts and the candidates are denoted as

$$\tilde{w} = [w; F(w), f_{mcan}(w)],$$

where $F(w)$ is a vector about knowledge-grounded features as specified in the task description section.

Dynamic Pooling Recurrent Networks

To encode contextual information, a dynamic pooling recurrent network is proposed, which contains two layers of recurrent units and one dynamic pooling layer between the two recurrent layers. In our model, a bi-directional LSTM 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 i th sequence ($i = 1, 2, \dots, l$) is encoded as

$$\begin{aligned}
 \vec{h}_{i,t}^1, \vec{c}_{i,t}^1 &= \overrightarrow{\text{LSTM}}^1(\vec{w}_{i,t-1}^U, \vec{h}_{i,t-1}^1, \vec{c}_{i,t}^1), \\
 \overleftarrow{h}_{i,t}^1, \overleftarrow{c}_{i,t}^1 &= \overleftarrow{\text{LSTM}}^1(\vec{w}_{i,t+1}^U, \overleftarrow{h}_{i,t+1}^1, \overleftarrow{c}_{i,t}^1),
 \end{aligned}$$

where h and c are the hidden states and cell states respectively. Different from HRED, which encodes each utterance separately, the first layer of Dynamic Pooling LSTM Networks encodes utterances by concatenating the utterances as a single sequence. Therefore, the initial LSTM hidden state of an utterance is the last hidden state from the encoded previous sequence:

$$\begin{aligned}
 \vec{h}_{i,0}^1, \vec{c}_{i,t}^1 &= \vec{h}_{i-1,n_i}^1, \vec{c}_{i-1,n_i}^1 \\
 \overleftarrow{h}_{i,0}^1, \overleftarrow{c}_{i,t}^1 &= \overleftarrow{h}_{i+1,n_i}^1, \overleftarrow{c}_{i+1,n_i}^1
 \end{aligned}$$

In the second part of the dynamic pooling recurrent network, the dynamic pooling layer is used to generate one vector representation \hat{h}_i^1 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:

$$\hat{h}_i^1 = \max([\vec{h}_{i,1}^1; \overleftarrow{h}_{i,1}^1], [\vec{h}_{i,2}^1; \overleftarrow{h}_{i,2}^1], \dots, [\vec{h}_{i,n_i}^1; \overleftarrow{h}_{i,n_i}^1]),$$

where $\max(\cdot)$ is the operation of max pooling of the vectors over dimensions. Finally, there is another bi-directional LSTM layer, which encodes utterance-level representations

$$\begin{aligned}
 \vec{h}_i^2, \vec{c}_i^2 &= \overrightarrow{\text{LSTM}}^2(\hat{h}_i^1, \hat{h}_{i-1}^2, \vec{c}_{i-1}^2), \\
 \overleftarrow{h}_i^2, \overleftarrow{c}_i^2 &= \overleftarrow{\text{LSTM}}^2(\hat{h}_i^1, \hat{h}_{i+1}^2, \vec{c}_{i+1}^2),
 \end{aligned}$$

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

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

Candidate Selection

Each candidate x_j is encoded by the first LSTM layer in the Dynamic Pooling LSTM:

$$\begin{aligned} \vec{h}_{j,t}^x, \vec{c}_{j,t}^x &= \overrightarrow{\text{LSTM}}^1(\vec{w}_{j,t-1}^U, \vec{h}_{j,t-1}^x, \vec{c}_{j,t}^x), \\ \overleftarrow{h}_{j,t}^x, \overleftarrow{c}_{j,t}^x &= \overleftarrow{\text{LSTM}}^1(\vec{w}_{j,t+1}^U, \overleftarrow{h}_{j,t+1}^x, \overleftarrow{c}_{j,t}^x), \end{aligned}$$

and max pooling is applied over the outputs to get the representation of the candidate:

$$r_j^X = \max([\vec{c}_{j,1}^x; \overleftarrow{c}_{j,1}^x], [\vec{c}_{j,2}^x; \overleftarrow{c}_{j,2}^x], \dots, [\vec{c}_{j,m_j}^x; \overleftarrow{c}_{j,m_j}^x]).$$

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

$$p(x_j) = \sigma(H_1(H_2([r^C; r_j^X; r^C \odot r_j^X; r^C - r_j^X])),$$

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

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

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.

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 is between 1 and 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.

Baseline Systems

- Dual Encoder (Lowe et al. 2015): uses two LSTMs with tied weights to encode the context $d = \{u_1, u_2, \dots, u_l\}$ 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.

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). The hidden layer size of LSTM is 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.

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 LSTM Our dynamic pooling 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 mar-

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

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	64.86	72.93	3	80.4	49.14	64.77	61	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 2: The official testing results of our submitted systems. Two different test sets for advising 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.

gin on the ubuntu dataset, yielding a 23.5% relative improvement. It demonstrates that the MCAN feature $f_{mcan}(w)$ also helps our model achieve slightly 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 dynamic pooling LSTM with both auxiliary features added to the input word representations (row (f)), achieving about 66% and 60% average performance for ubuntu and advising datasets respectively.

Official Evaluation

In the DSTC7 challenge, the proposed systems are submitted for official evaluation. 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, and Table 3 shows the final rankings of our results for all subtasks. Note that the model for the subtask 5 is the single model, which performs worse than the ensemble one as the subtask 1. In the official evaluation, the superior performance and the achieved rankings across different subtasks significantly demonstrate the effectiveness of the proposed model. Considering that our rankings are either

	Ubuntu	Advising
Subtask 1	3	2
Subtask 3	-	3
Subtask 4	2	3
Subtask 5	6	4
Overall	4	3

Table 3: Official rankings of our systems on each subtask. The overall ranking considers all subtasks.

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

Table 4: Ablation results on Ubuntu development set (%).

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.

Ablation Study

To further understand the contribution of each component, we conduct an ablation test on the RAP-Net model. Table 4 shows the ablation results on the ubuntu subtask 1 development set. We remove one component at each time 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. This set-

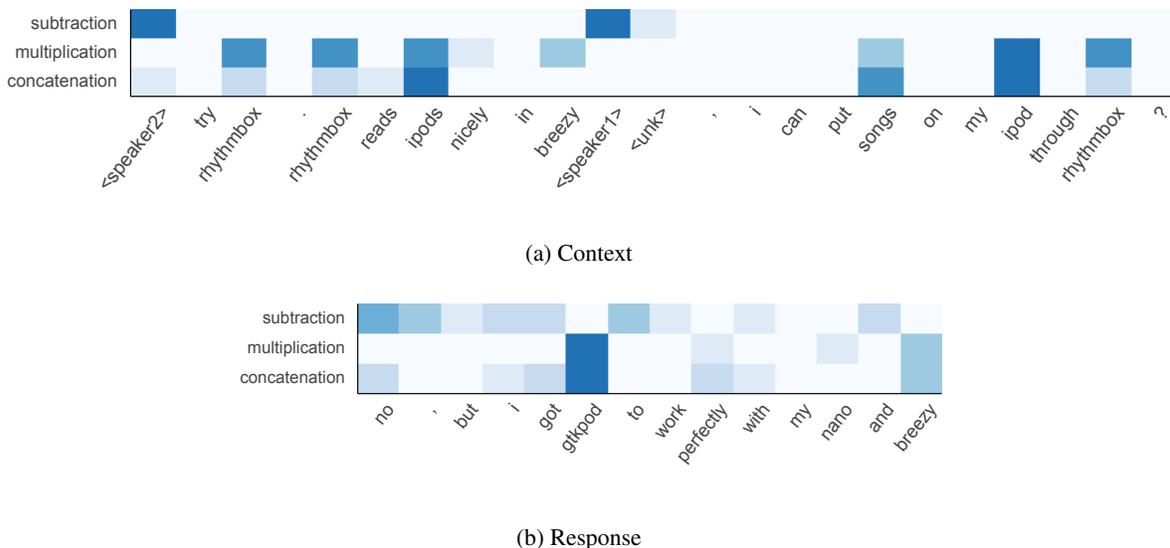


Figure 4: Visualization of attention scores. We plot the attention scores f_{mcan} of two sequences from Ubuntu development set: (a) A partial conversation and (b) The correct response corresponding to the conversation. The conversation is truncated to the last two sentences due to width limitation.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

ting is equivalent to HRED with an additional feature f_{mcan} .

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 each of them results in a reduction of roughly 1 in terms of the average score, so the highway encoder and the dynamic pooling also slightly contribute to the improvement. Furthermore, the intra-attention benefits least to performance, which is similar to the findings in the prior work (Tay, Tuan, and Hui 2018).

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 Figure 4. It can be found that the features extracted by mean pooling, max pooling and intra-attention are always equal or close to zero 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 Figure 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 larger 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 largest 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.

Conclusion

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.

References

- Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- 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.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kummerfeld, J. K.; Gouravajhala, S. R.; Peper, J.; Athreya, V.; Gunasekara, C.; Ganhotra, J.; Patel, S. S.; Polymenakos, L.; and Lasecki, W. S. 2018. Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *arXiv preprint arXiv:1810.11118*.
- 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*.
- Mikolov, T.; Grave, E.; Bojanowski, P.; Puhersch, C.; and 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., and Thyagarajan, A. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- 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*.
- Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.
- Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, 3776–3784.
- 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.
- Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- 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*.
- 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*.