# Learning to Detect Relevant Contexts and Knowledge for Response Selection in Retrieval-based Dialogue Systems

Kai Hua*
Computer Center, Peking University
kifish@pku.edu.cn

Zhiyuan Feng*
Department of Computer Science and Technology, Peking University
fengzhiyuan@pku.edu.cn

Chongyang Tao
Wangxuan Institute of Computer Technology , Peking University
chongyangtao@pku.edu.cn

Rui Yan†
Wangxuan Institute of Computer Technology, Peking University
Beijing Academy of Artificial Intelligence (BAAI)
ruiyan@pku.edu.cn

Lu Zhang†
Department of Computer Science and Technology, Peking University
zhanglucs@pku.edu.cn

## ABSTRACT

Recently, knowledge-grounded conversations in the open domain gain great attention from researchers. Existing works on retrieval-based dialogue systems have paid tremendous efforts to utilize neural networks to build a matching model, where all of the context and knowledge contents are used to match the response candidate with various representation methods. Actually, different parts of the context and knowledge are differentially important for recognizing the proper response candidate, as many utterances are useless due to the topic shift. Those excessive useless information in the context and knowledge can influence the matching process and leads to inferior performance. To address this problem, we propose a multi-turn **R**esponse **S**election **M**odel that can **D**etect the relevant parts of the **C**ontext and **K**nowledge collection (**RSM-DCK**). Our model first uses the recent context as a query to pre-select relevant parts of the context and knowledge collection at the word-level and utterance-level semantics. Further, the response candidate interacts with the selected context and knowledge collection respectively. In the end, The fused representation of the context and response candidate is utilized to post-select the relevant parts of the knowledge collection more confidently for matching. We test our proposed model on two benchmark datasets. Evaluation results indicate that our model achieves better performance than the existing methods, and can effectively detect the relevant context and knowledge for response selection.

*Equal contribution.
†Corresponding authors: Rui Yan (ruiyan@pku.edu.cn) and Lu Zhang (zhanglucs@pku.edu.cn) .


*CIKM '20, October 19–23, 2020, Virtual Event, Ireland*

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411967

## CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**;

## KEYWORDS

deep neural network, matching, multi-turn response selection, retrieval-based conversation, knowledge-grounded conversation

**ACM Reference Format:**
Kai Hua, Zhiyuan Feng, Chongyang Tao, Rui Yan, and Lu Zhang. 2020. Learning to Detect Relevant Contexts and Knowledge for Response Selection in Retrieval-based Dialogue Systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411967

## 1 INTRODUCTION

The human-machine conversation is the ultimate goal of artificial intelligence. Recently, building a conversation system with intelligence has drawn increasing interest in academia and industry. Existing studies can be generally categorized into two groups. The first group is retrieval-based dialogue systems [8, 30, 37, 39, 52] which select the proper response from the response candidates under the given user input or dialogue context, and have been applied in many industrial products such as XiaoIce from Microsoft [28] and AliMe Assist from Alibaba [13]. The second group is generation-based dialogue systems [14, 26, 27] which generate the response word by word under an encoder-decoder framework [26, 27]. In this paper, we focus on response selection tasks in the retrieval-based dialogues.

The key to the traditional response selection task is to measure the matching degree between the dialogue context and response candidate. Early studies focus on the single-turn dialogue systems with the user input as the query [10, 32, 33], and recently shift to the multi-turn dialogue systems with both the dialogue context and the user input considered, aiming at ranking response candidates by calculating semantic relevance between the dialogue context and response candidates. The sequential matching network [36] proposes a representation-matching-aggregation framework which is followed by subsequent work. The model conducts matching between utterances and the response candidate on the word and

**Table 1: An example of document-grounded dialogue from Persona-Chat dataset.**

| | |
|---|---|
| **A's profile** | horror movies are my favorites.<br>i am a stay at home dad.<br>my father used to work for home depot.<br>i spent a decade working in the human services field.<br>i have a son who is in junior high school. |
| **B's profile** | i read twenty books a year.<br>i am a stunt double as my second job.<br>i only eat kosher.<br>i was raised in a single parent household. |
| **Context** | **A**: hello what are doing today?<br>**B**: i am good, i just got off work and tired, i have two jobs.<br>**A**: i just got done watching a horror movie<br>**B**: i rather read, i have read about 20 books this year.<br>**A**: wow! i do love a good horror movie. loving this cooler weather<br>**B**: but a good movie is always good.<br>**A**: yes! my son is in junior high and i just started letting him watch them too |
| **True response** | i work in the movies as well. |
| **False response** | that is great! are you going to college ? |

utterance level, and aggregates matching features to obtain the matching score. The deep attention matching network [52] captures matching features on multiple levels of granularity. Interaction over interaction network [30] aggregates matching features and supervises those at each level of granularity directly.

Human conversations tend to be related to background knowledge. For instance, when two persons talk about a movie, information about the movie is already some parts of the prior knowledge in their brains. Based on this observation, it is necessary for chatbots to be grounded in external knowledge, which can make human-machine conversations more practical and engaging [4]. Therefore, the highlight of research in the dialogue systems has shifted to incorporating the external knowledge into chatbots recently. The Starspace network [35] learns the task-specific embedding and selects the appropriate response by the cosine similarity between the dialogue context concatenated by the associated document and the response candidate. The profile memory network [43] uses the dialogue context as the query and performs attention over the document to combine with the query, and then measures the similarity between the fused query and response candidate. The document grounded matching network [46] makes the dialogue context and document interact with each other and interact via the response candidate individually later with the hierarchical attention mechanism. Dual-interaction-matching-network [7] makes the dialogue context and document interact with the response candidate respectively via the cross-attention mechanism.

Existing work on retrieval-based dialogue systems has paid tremendous efforts to utilize the neural network to build various text-matching model, in which all of the context and knowledge contents are used to match the response candidate. Actually, some parts of the context and knowledge collection are irrelevant to the response candidate, which may introduce some noise confusing the model, thus leading to inferior performance. To illustrate the problem, we present an example in table 1. In the recent dialogue context, the current topic is about movies, so the previous utterances about movies should be given more weight, and utterances that are unrelated to movies (e.g. the fourth utterance of the dialogue context) should be given less weight. Similarly, in the knowledge collection, for instance, the second sentence of the B's profile should be assigned more weight than other profile sentences. This example indicates that different parts of the context and knowledge are differentially important for recognizing the proper response candidate, as many utterances are useless. Those excessive useless information in the context and knowledge can influence the matching process.

To address the above problem, we propose a multi-turn **R**esponse **S**election **M**odel that can **D**etect the relevant parts of the **C**ontext and **K**nowledge collection (**RSM-DCK**). For simplicity of expression, we refer to the knowledge collection as documents, however, RSM-DCK can be adapted to other types of knowledge resources such as knowledge graphs. The model first uses the recent utterance of the dialogue context as a query to pre-select relevant parts of the dialogue context and document. Then the selected dialogue context and document interact with the response candidate individually by cross-attention, and an BiLSTM [5, 9] is employed to aggregate the matching features of the context, document, and response candidate respectively. Due to the inter-dependency and temporal relationship among utterances in the dialogue context, another BiLSTM is adopted to accumulate the dialogue context. To select the most relevant sentences in the knowledge collection, the fused representation of the pre-selected dialogue context and response candidate is utilized as the query to post-select the document with the attention mechanism for the reason that sentences in the document are relatively independent and the true response tends to be solely related to one of them.

We conduct experiments on two benchmark datasets for multi-turn knowledge-grounded response selection: the Original/Revised Persona-Chat dataset [43], and the CMUDoG dataset [50]. The evaluation results indicate that our model outperforms previous models in terms of all metrics on all the datasets. Compared with dual-interaction-matching-network, the best performing baseline on all the two benchmarks, our model achieves 0.9% absolute improvement on $R_{20}$@1 on the Original Persona-Chat dataset, 1.2% absolute improvement on $R_{20}$@1 on the Revised Persona-Chat dataset, and 0.7% absolute improvement on $R_{20}$@1 on the CMUDoG dataset.

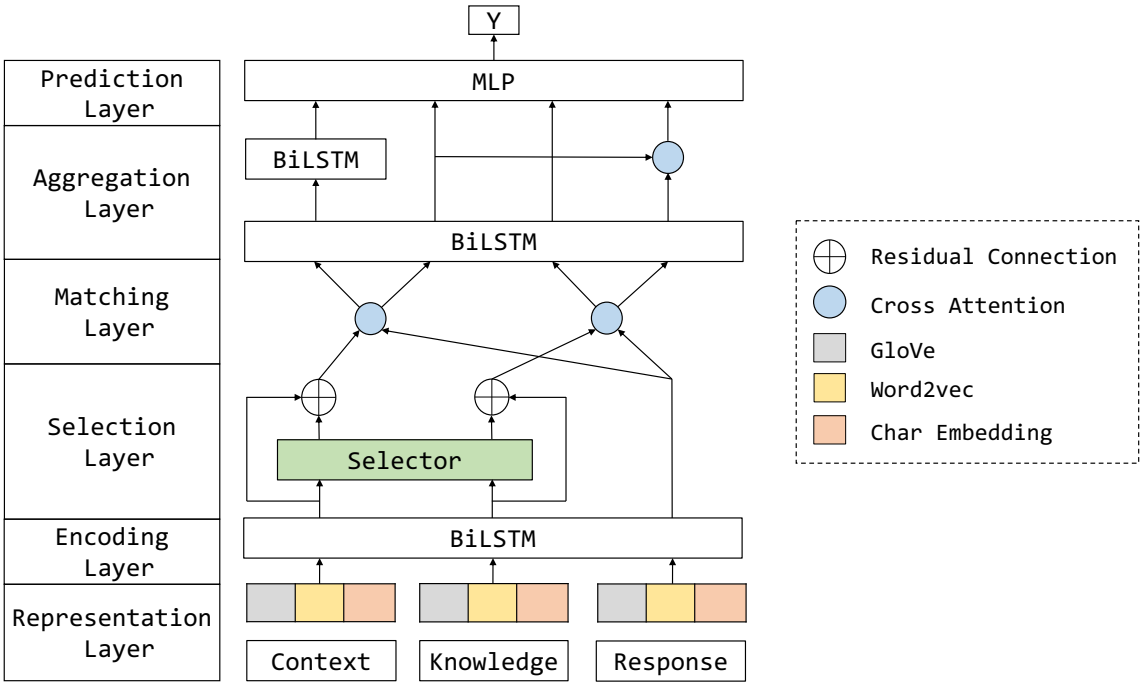


**Figure 1: Architecture of the proposed RSM-DCK.**

## 2 RELATED WORK

Retrieval-based open-domain dialogue systems learn a matching model to calculate the similarity between the user input and the candidate response for response selection. Early studies focus on the single-turn dialogue with the user input regarded as the query [10, 32, 33], and recently move to the multi-turn dialogue where the dialogue history and the user input are both taken into account. Representative methods include the Dual LSTM [18], the deep learning-to-respond architecture [39], the multi-view matching model [51], the sequential matching network [36], the deep attention matching network [52], the multi-representation fusion network [29], the interaction-over-interaction matching network [30], and the interaction matching network [6].

Knowledge is crucial to the dialogue in the real world. The lack of knowledge makes dialogue systems suffer from semantics issues [49], consistency issues [15, 24, 47], and interactiveness issues [25, 34, 48]. Recently, incorporating knowledge into the dialogue has been shown beneficial [7, 46]. Existing work for knowledge-grounded dialogue systems can be generally categorized into two groups by the type of knowledge. The first group is document-grounded dialogue systems where the documents are unstructured texts such as user profiles and Wikipedia articles [4, 7, 43, 46, 50]. The second group is knowledge-graph-grounded dialogue systems where the knowledge graph is a large network of entities which contains semantic types, properties, and relationships between entities [17, 40, 49]. However, either of the groups can be influenced by irrelevant parts of the context and knowledge collection, thus leading to a severe performance drop.

Recently, Multi-turn dialogue context selection has been widely explored in the response generation and response selection. In the response generation, Xing et al. [38] propose a hierarchical recurrent attention network where the hidden state in the decoder is utilized to select important parts of the dialogue context on both the word- and utterance-level via the hierarchical attention mechanism. To effectively model the context of a dialogue, Zhang et al. [44] propose two types of attention mechanisms (including dynamic and static attention) to weigh the importance of each utterance in a conversation and obtain the contextual representation. Zhang et al. [42] propose a model named ReCoSa where the self-attention mechanism is employed to update both the context and masked response representation, and the attention weights between each context and response representations are computed as the relevant score to guide the decoding process. In the response selection, Zhang et al. [45] propose a deep utterance aggregation model where the last utterance representation refines the preceding utterances to obtain the turns-aware representation and then the self-attention mechanism is applied among utterances. Yuan et al. [41] propose a multi-hop selector network where the latter parts of the dialogue context are used as the query to select the relevant utterances on both the word- and utterance-level. As the knowledge sometimes contains a lot of redundant entries in knowledge-grounded conversation, Lian et at. [16] propose a model with the knowledge selection mechanism which leverages both prior and posterior distributions over the knowledge to facilitate knowledge selection. Kim et al. [11] propose a sequential knowledge transformer that employs a sequential latent variable model to better leverage the response information for the proper choice of the knowledge collection in multi-turn dialogue. Different from previous work, our

model both selects the dialogue context and knowledge collection with the carefully designed selection mechanism.

## 3 PROBLEM FORMALIZATION

Suppose that we have a dialogue dataset with knowledge collection $\mathcal{D} = \{c_i, k_i, r_i, y_i\}_{i=1}^{N}$, where $c_i = \{u_{i,1}, u_{i,2}, ..., u_{i,m_i}\}$ represents a conversational context and $m_i$ is the number of utterances in the conversational context; $k_i = \{k_{i,1}, k_{i,2}, ..., k_{i,n_i}\}$ represents a collection of knowledge and $n_i$ is the number of sentences in the collection of knowledge; $r_i$ denotes a candidate response; and $y_i \in \{0, 1\}$ is the label. $y_i = 1$ indicates that $r_i$ is a proper response for $c_i$ and $k_i$, otherwise, $y_i = 0$. $N$ is the number of training samples. The task is to learn a matching model $g(c, k, r)$ from $\mathcal{D}$, and thus for a new context-knowledge-response triple $(c, k, r)$, $g(c, k, r)$ returns the matching degree between $r$ and $(c, k)$.

## 4 MODEL

### 4.1 Model Overview

We propose a matching network with deep interaction to model $g(c, k, r)$. Figure 1 shows the architecture of RSM-DCK, which consists of six layers, namely the representation layer, encoding layer, selection layer, matching layer, aggregation layer, and predication layer. In the representation layer, multiple types of embedding are adopted to represent words. In the encoding layer, an BiLSTM is adopted to encode the context, knowledge collection, and candidate responses. In the selection layer, the recent parts of the context are used as the query to pre-select the relevant parts of the context $c$ and knowledge collection $k$. In the matching layer, the context $c$ and knowledge collection $k$ interacts with the candidate response $r$ respectively by the attention mechanism. In the aggregation layer, another BiLSTM is adopted to extract matching signals considering the dependency among utterances in the context. As shown in table 1, sentences in the knowledge collection might be independent of each other. Therefore, matching information of the knowledge collection is aggregated by attending to the context and response candidate, which can be regarded as the post-selection. In the end, the matching feature is fed into the prediction layer to measure the matching degree between the candidate response $r$ and the context-knowledge pair $(c, k)$.

### 4.2 Representation Layer

We use general pre-trained word embedding, namely GloVe [23] to represent each word of the context, knowledge, and the response candidate. To alleviate the out-of-vocabulary issue, Word2vec [20] trained on the task-specific training set and character-level embedding is adopted in the representation layer. We simply concatenate the three embeddings for each word. Therefore, the representation layer is capable to capture both semantics and morphology of words.

Formally, given an utterance $u_i$ in a context $c$, a sentence $k_i$ in the knowledge collection and a response candidate $r$, we embed $u_i$, $k_i$, and $r$ as $\mathbf{E}_{u_i} = [\mathbf{e}_{u_{i,1}}, \mathbf{e}_{u_{i,2}}, ..., \mathbf{e}_{u_{i,l_{p_i}}}]$, $\mathbf{E}_{k_i} = [\mathbf{e}_{k_{i,1}}, \mathbf{e}_{k_{i,2}}, ..., \mathbf{e}_{k_{i,l_{q_i}}}]$ and $\mathbf{E}_r = [\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, ..., \mathbf{e}_{r_{l_r}}]$ respectively, where $\mathbf{e}_{u_{i,t}}$, $\mathbf{e}_{k_{i,t}}$ and $\mathbf{e}_{r_t}$ are a $d$-dimension vector concatenated corresponding to the $t$-th word in the $u_i$, $k_i$, $r$ respectively, and $l_{p_i}$, $l_{q_i}$, and $l_r$ are the number of words in the sequences respectively.

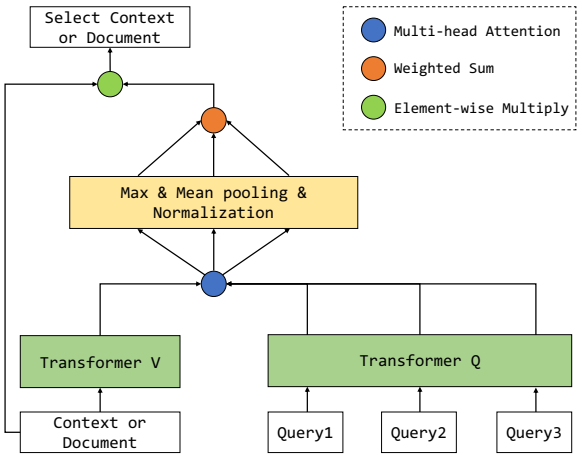


**Figure 2: Architecture of the selector. For ease of illustration, we draw three representations for each utterance.**

### 4.3 Encoding Layer

We employ an BiLSTM to model the bidirectional interactions among words in the utterances, and encode each utterance as a sequence of hidden vectors. The sentence encoding can be denoted as follows,

$$u_{i,j} = \texttt{BiLSTM}_1(\mathbf{E}_{u_i}, j), j \in \{1, 2, ..., l_{p_i}\} \quad (1)$$

$$k_{i,j} = \texttt{BiLSTM}_1(\mathbf{E}_{k_i}, j), j \in \{1, 2, ..., l_{q_i}\} \quad (2)$$

$$r_j = \texttt{BiLSTM}_1(\mathbf{E}_r, j), j \in \{1, 2, ..., l_r\} \quad (3)$$

where $u_{i,j}$ is the encoded representation of the $j$-th word in the $i$-th context utterance, $k_{i,j}$ is the encoded representation of the $j$-th word in the $i$-th knowledge sentence, and $r_j$ is the encoded representation of the $j$-th word in the response candidate. LSTM can be utilized to encode a sequence of input vectors $(w_1, w_2, ..., w_n)$ into hidden vectors $(h_1, h_2, ..., h_n)$, which can be formulated as follows,

$$i_t = \sigma(U_i w_t + V_i h_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(U_f w_t + V_f h_{t-1} + b_f) \quad (5)$$

$$o_t = \sigma(U_o w_t + V_o h_{t-1} + o_f) \quad (6)$$

$$g_t = tanh(U_g w_t + V_g h_{t-1} + b_g) \quad (7)$$

$$C_t = i_t \circ g_t + f_t \circ C_{t-1} \quad (8)$$

$$h_t = o_t \circ tanh(C_t) \quad (9)$$

where "$\circ$" denotes element-wise product. BiLSTM places one LSTM in the forward direction and another in the backward direction. We concatenates the outputs of the two LSTMs, which is defined as:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (10)$$

### 4.4 Selection Layer

Figure 2 depicts the content selector module in RSM-DCK. Following the MRFN [29], we use the Attention Module. Three input sentences, namely the query sentence $\mathcal{Q}$, the key sentence $\mathcal{K}$, and

the value sentence $\mathcal{V}$, are fed into the Attention Module. To be specific, $\mathcal{Q} = \{e_i\}_{i=1}^{n_q}$, $\mathcal{K} = \{e_i\}_{i=1}^{n_k}$,$\mathcal{V} = \{e_i\}_{i=1}^{n_v}$, where $n_q$,$n_k$, and $n_v$ denote the number of words in each sentence. And $e_i$ is the feature vector of the $i$-th word in the corresponding sequence. The Attention Module uses Scaled Dot-Product Attention [31] to model the attention mechanism between the $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$, which can be defined as:

$$\texttt{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{softmax}(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d}})\mathcal{V} \tag{11}$$

Intuitively, each word of $\mathcal{V}$ is weighted by an importance score calculated by the similarity between a word in $\mathcal{Q}$ and a word in $\mathcal{K}$. By equation 11, we can obtain $\mathcal{O}$, a new representation of $\mathcal{Q}$, in which the new feature vector of each word in $\mathcal{Q}$ is a weighted sum of the representation of words in $\mathcal{V}$. To improve the expressiveness of the module, a linear transformation operation is adopted before the attention operation. A layer normalization operation [1] is applied to $\mathcal{O}$, which helps ease vanishing or exploding of gradients. Then a feed-forward network FFN with RELU is applied upon the normalization result, which helps further enhance the expressiveness of the module. A residual connection is adopted to help gradient flow. Finally, the multi-head mechanism is applied to help the attention module capture diverse features of the input.

In RSM-DCK, the last utterance of the context is used as a query $q$ (namely $u_{l_{p_i}}$) for the reason that the last utterance is usually the most related utterance with the response.

*4.4.1 Context selector.* To capture long-term dependency in the sequences, we use self-attention layer to further transform the query $q$ and each utterance $u_i$ in the context, which can be formulated as follows,

$$\hat{Q} = \texttt{Attention}(Q, Q, Q) \tag{12}$$

$$\hat{U}_i = \texttt{Attention}(U_i, U_i, U_i) \tag{13}$$

where $Q \in \mathbb{R}^{l_{p_i} \times 2d}$ is the representation matrix of $u_{n_c}$ produced by the encoding layer with $l_q$ is the number of words in the query sentence; $U_i \in \mathbb{R}^{l_{p_i} \times 2d}$ denotes the representation matrix of $u_i$ and $k_i$ produced by the encoding layer.

To pre-select the relevant parts of the context, we use the scaled dot-product attention to calculate the similarity matrix between the query utterance and the previous utterances in the context, which is defined as:

$$\phi_i = \frac{\hat{Q} \cdot \hat{U}_i^T}{\sqrt{2d}} \tag{14}$$

where $\phi_i \in \mathbb{R}^{l_q \times l_{p_i}}$. The operation is applied to each utterance in the context, thus we obtain $\Phi \in \mathbb{R}^{l_q \times n_c \times l_{p_i}}$, where $n_c$ is the number of utterances in the context. From the perspective of the query, a max pooling operation is adopted to select the most related context:

$$\tilde{\Phi} = \max_{dim=0} \Phi \tag{15}$$

where $\tilde{\Phi} \in \mathbb{R}^{n_c \times l_{p_i}}$

To obtain the relevant score between the query and each utterance in the context, we perform max pooling and mean pooling operation over $\tilde{\Phi}$, so as to summarizes the maximum value and average value of the attention weight, formulated as:

$$s_1 = \max_{dim=1} \tilde{\Phi}, \tag{16}$$

$$s_2 = \underset{dim=1}{mean}\, \tilde{\Phi} \tag{17}$$

where $s_1, s_2 \in \mathbb{R}^{n_c}$. We then use a hype-parameter $\alpha$ to fuse the two scores:

$$s = \alpha \cdot s_1 + (1 - \alpha) \cdot s_2 \tag{18}$$

$$s = \texttt{softmax}(s) \tag{19}$$

We set initial value of $\alpha$ to 0.5 and make it update during the training.

Following the MSN [41], we also perform multi-hop selections to enhance the robustness of the content selection module. We first concatenate the representation of last $m$ utterances in the context $\{u_{n_c-m+1}, u_{n_c-m+2}, \cdots, u_{n_c}\}$, and then carry out a mean-pooling operation to obtain the new query representation $Q^{(m)}$. We use the new query $q$ to conduct the above selection process, thus obtain $m$ different weight distributions $s_c = [s^{(1)}, s^{(2)}, \cdots, s^{(m)}]$. To fuse the weight distributions across different hops, we utilize an adaptive combination to compute the final weight distribution $\bar{s}_c \in \mathbb{R}^{n_c \times 1}$ as follows,

$$\bar{s}_c = s_c \cdot \pi \tag{20}$$

where $\pi \in \mathbb{R}^{k \times 1}$ is a learnt weight.

Finally, we can obtain the updated representation $\bar{U}_i$ for $i$-th utterance ($u_i$) in the context, which can be formulated as follows,

$$\bar{U}_i = U_i + \bar{s}_c(i) * U_i \tag{21}$$

where $\bar{s}_c(i)$ is the relevance score for $i$-th utterance. The the residual connection is used here to prevent the gradient vanishing.

*4.4.2 Knowledge selector.* Similarly, we also employ a self-attention layer to process each utterance $k_i$ in the knowledge, which can be formulated as follows,

$$\hat{K}_i = \texttt{Attention}(K_i, K_i, K_i) \tag{22}$$

where $K_i \in \mathbb{R}^{l_{q_i} \times 2d}$ represents the representation matrix of $k_i$ produced by the encoding layer.

Then we can calculate the final weight distribution $\bar{s}_k$ and the updated knowledge representation $\bar{K}_i$ by replacing the $\hat{U}_i$ with $\hat{K}_i$ in the formulation of context selector.

$$\bar{K}_i = K_i + \bar{s}_k(i) * K_i \tag{23}$$

To conclude, we use the latter parts of the context to pre-select relevant parts of the context and knowledge collection in the selection layer, which helps RSM-DCK focus on the relevant contents in the following matching and aggregation stage. In addition, RSM-DCK is capable of obtaining the explicit weight distribution indicating the relative importance of each utterance in the context or each sentence in the knowledge collection.

## 4.5 Matching Layer

Following the ESIM [2], we use the cross attention mechanism to model the interaction between context and the candidate response, and the interaction between the knowledge collection and the candidate response. The context with multi-turn utterances is concatenated as a long sequence, which can be formally expressed as

$C = [U_1, U_2, \cdots, U_{n_c}] \in \mathbb{R}^{l_c \times 2d}$ with $l_c$ the total number of words in the context (namely $l_c = \sum_{i=1}^{n_c} l_{p_i}$ ). We then use dot product to calculate the similarity between the $i$-th word in the context $c$ and the $j$-th word in the candidate response $r$, which can be formulated as follows,

$$\mathcal{E} = C^T \cdot R \tag{24}$$

where $R \in \mathbb{R}^{l_r \times 2d}$ denotes the representation matrix of $r$ produced by the encoding layer; $\mathcal{E} \in \mathbb{R}^{l_c \times l_r}$ is the attention matrix. We can obtain the weight matrices $\alpha \in \mathbb{R}^{l_c \times l_r}$ and $\beta \in \mathbb{R}^{l_c \times l_r}$ by normalization alongside the column and the row respectively, which can be formulated as follows,

$$\alpha_{i,j} = \frac{\exp(\mathcal{E}_{i,j})}{\sum_{j=1}^{l_r} \exp(\mathcal{E}_{i,j})} \tag{25}$$

$$\beta_{i,j} = \frac{\exp(\mathcal{E}_{i,j})}{\sum_{i=1}^{l_c} \exp(\mathcal{E}_{i,j})} \tag{26}$$

Through a weighted combination, we can obtain a response-aware context representation $\hat{C}$ and a context-aware response representation $\hat{R}$ for the context and the response candidate respectively, formulated as:

$$\hat{C} = \alpha \cdot R \tag{27}$$

$$\hat{R} = C \cdot \beta \tag{28}$$

Referring to TBCNN [21], to measure the aligned token-level semantics similarity, we introduce the following functions to obtain the matching features $\bar{C}$ and $\bar{R}$,

$$\bar{C} = [C, \hat{C}, C - \hat{C}, C \circ \hat{C}] \tag{29}$$

$$\bar{R} = [R, \hat{R}, R - \hat{R}, R \circ \hat{R}] \tag{30}$$

where "$\circ$" denotes element-wise product operation.

Similarly, we also concatenated all knowledge entries as a long sequence $K^c$, and then perform the above cross attention, thus obtaining a response-aware knowledge representation $\hat{K}^c$ and a knowledge-aware response representation $\hat{R}^c$ respectively. The two representations are further transformed into the matching features $\bar{K}^c$ and $\bar{R}^c$ via the following formulations,

$$\bar{K}^c = [K^c, \hat{K}^c, K^c - \hat{K}^c, K^c \circ \hat{K}^c] \tag{31}$$

$$\bar{R}^c = [R, \hat{R}^c, R - \hat{R}^c, R \circ \hat{R}^c] \tag{32}$$

## 4.6 Aggregation Layer

In the aggregation layer, matching information is first aggregated on the sentence-level. We convert the representation of the context $\bar{C}$ back to the representations of each utterance as $\{\ddot{U}_i\}_{i=1}^{n_c}$. Then, an BiLSTM is adopted, followed by max pooling and last hidden state pooling to obtain a fixed-length vector for each utterance in the context. The process can be formally expressed as follows,

$$\ddot{U}_{i,j} = \texttt{BiLSTM}_2(\ddot{U}_i, j), j \in \{1, 2, \cdots, l_{p_i}\} \tag{33}$$

$$U_i^s = [\texttt{max}(\ddot{U}_i); \ddot{U}_{i,l_{p_i}}] \tag{34}$$

where $U_i^s$ denotes the matching feature between the $i$-th utterance and the response candidate. We also process the context-aware response representation $\bar{R}$ and the knowledge-aware response representation $\bar{R}^c$ with the same operation described in Equation (33-34), and obtain aggregated features $M_r$ and $M_r^c$ respectively.

To aggregate the matching information in the session-level, we adopt another BiLSTM (denoted as $\texttt{BiLSTM}_3$) to process a sequence of matching feature $\{U_i^s\}_{i=1}^{n_c}$, and take the concatenation of the final hidden vector and the max-pooling of the hidden vectors as the final matching feature between the context and the response candidate, denoted as $M_c$.

For the knowledge $\bar{K}^c$, similarly, we first convert the representation into the representations of each sentence $\{\ddot{K}_i^c\}_{i=1}^{n_k}$. Then we process each sentence in the knowledge collection with the formulation in Equation (33-34), and obtain the matching feature between $i$-th sentence in the knowledge collection and the response candidate, denoted as $K_i^s$. Different from the diglogue context, sentences in the knowledge collection are relative independent to each other, so we aggregate the matching information of the knowledge collection via the attention mechanism, which can be regarded as the post-selection for knowledge. To be specific, we use $M_r^c$ which contains high-level information about the dialogue context and candidate response to attend to $K^s$, thus obtaining $M_k$. The process is defined as:

$$\gamma_i = \frac{\exp(K_i^s \cdot M_r^c)}{\sum_{i=1}^{n_k} \exp(K_i^s \cdot M_r^c)} \tag{35}$$

$$M_k = \sum_{i=1}^{n_k} \gamma_i K_i^s \tag{36}$$

Finally, we take the concatenation of $M_c$, $M_k$, $M_r$, and $M_r^c$ as the final matching feature, formulated as:

$$M_{\texttt{final}} = [M_c; M_k; M_r; M_r^c] \tag{37}$$

## 4.7 Prediction Layer

The final matching feature vector $M_{\texttt{final}}$ is then fed into a multi-layer perceptron (MLP) classifier to get the matching score between the context, the knowledge and the candidate response, which is defined as:

$$g(c, k, r) = f_2(W_2^\top \cdot f_1(W_1^\top \cdot M_{\texttt{final}} + b_1) + b_2), \tag{38}$$

where $W_1, W_2, b_1$, and $b_2$ are parameters. $f_1(\cdot)$ is the tanh activation function, and $f_2(\cdot)$ is the softmax function.

## 4.8 Learning Method

As is shown in Equation (38), we adopt a softmax normalization layer over the matching logits of all candidate responses to speed up training and alleviate the imbalance of samples. Parameters of the model are optimized by minimizing the cross-entropy loss on $D$, which is defined as

$$\mathcal{L}(\Theta) = -\sum_{i=1}^{N} y_i \log g(c_i, k_i, r_i) \tag{39}$$

where $\Theta$ represents the parameters of the model.

# 5 EXPERIMENTS

## 5.1 Datasets and Evaluation

We test RSM-DCK on two benchmark datasets, namely the Persona-Chat dataset [43] and the CMUDoG dataset [50]. Statistics of the two datasets are shown in table 2.

**Table 2: Statistics of the two datasets.**

| Statistics | Persona-Chat | | | CMUDoG | | |
|---|---|---|---|---|---|---|
| | Train | Val | Test | Train | Val | Test |
| # of conversations | 8939 | 1000 | 968 | 2881 | 196 | 537 |
| # of turns | 65719 | 7801 | 7512 | 36159 | 2425 | 6637 |
| Av_turns / conversation | 7.35 | 7.80 | 7.76 | 12.55 | 12.37 | 12.36 |
| Av_length of utterance | 11.67 | 11.94 | 11.79 | 18.64 | 20.06 | 18.11 |

***Persona-Chat***. The first dataset we use is the Persona-Chat [43], which consists of 151,157 turns of conversations. The dataset is split as a training set, a validation set and a testing set by the publishers. In all the three sets, each utterance is associated with one positive response comes from the groud-truth and 19 negative response candidates that are randomly sampled from the dataset. To create the dataset, crowd workers are randomly paired and each of them is required to chat naturally with his partner according to his assigned profiles. The persona profiles describe the characteristics of the speaker, thus can be regarded as a collection of knowledge which consists of 4.49 sentences on average. For each dialogue, there are two format personas, namely, original profiles and revised profiles. The latter is rephrased from the former, which helps prevent models utilizing word overlap and makes response selection more challenging.

***CMUDoG***. Another dataset we use is the CMUDoG dataset published in [50]. The conversations about some certain documents are collected through Amazon Mechanical Turk. The topic of the documents is all about movies, thus helping interlocutors have a common topic to talk about naturally. To impel two paired workers to talk about the given documents, two scenarios are explored. In the first one, only one interlocutor can see the document while the other cannot. The interlocutor who has access to the given document is instructed to introduce the movie to the other. In the second one, both interlocutors have access to the given document and are required to talk about the contents of the document. Considering the small number of conversations in each scenario, data in the two scenarios is merged to form a larger dataset. Following the DGMN [46], we filter out conversations whose turns are less than 4 to avoid noise. Due to the lack of negative responses in the dataset, we sample 19 negative response candidates for each utterance from the same set.

Following previous studies [46, 50], we also employ $R_n@k$ as evaluation metrics, where $R_n@k$ ($n$=20, $k$=1,2,5) stands for recall at position $k$ in $n$ response candidates.

## 5.2 Baseline Models

We compare our model with the following models:

***IR Baseline*** *[43].* An information retrieval method which selects the appropriate response based on simple word overlap.

***Starspace*** *[35].* The model learns the task-specific embedding by minimizing the margin ranking loss and returns the cosine similarity between the conversation context concatenated by the associated document and the response candidate.

***Profile Memory*** *[43].* The model uses the memory network as the backbone. To be specific, the model uses the dialogue history as the query, and performs attention over the profile sentence to find the relevant lines to be combined with the query, and then measures the similarity between the fused query and response candidate.

***KV Profile Memory*** *[43]* . The model extends the Profile Memory to a multi-hop version. In the first hop, the model uses the Profile Memory to obtain the fused query. Then, in the second hop, the attention is conducted with the dialogue history as the key and the next dialogue utterance as the value. The output of the second hop is utilized to measure the similarity between itself and the response candidate.

***Transformer*** *[19]* . The model encodes the dialogue history and response candidate with Transformer encoder [31], and encodes the profile sentences via the bag-of-words representation instead.

***DGMN****[46].* The model encodes representations of the dialogue context and document with the self-attention, and fuses representations of the dialogue context and document into each other by the cross-attention, then interacts with the response candidate via the hierarchical attention individually.

***DIM****[7].* The model encodes representations of the dialogue context, document, and response candidate with the BiLSTM and lets the dialogue context and document interact with the response candidate respectively via the cross-attention mechanism. Finally, another BiLSTM is adopted to aggregate the matching features of the dialogue context, document, and response candidate. This model outperforms all baselines above.

## 5.3 Implementation Details

In RSM-DCK, we set the dimension of the glove embedding as 300, and word2vec is trained on the training set with the dimension of the word embedding set as 100. The dimension of the character-level embedding is set as 150 with window size {3, 4, 5}, each containing 50 filters. We freeze the embedding weights during training. The hidden size of all BiLSTMs is set as 300, and the number of head of the Attention Module is 3. The MLP in the prediction layer has 256 hidden units, which is activated by ReLU [22]. In the Persona-Chat dataset, the maximum number of utterances per dialogue context is set as 15, and 5 for sentences per document. We set the maximum number of words as 20 for each utterance in the dialogue context, each sentence in knowledge collection, and each candidate response (i.e., $l_{p_i} = l_{q_i} = l_r = 20$) . If the number of words is less than 20, we pad zeros, otherwise, we keep the latter 20 words. In the CMUDoG dataset, the maximum number of utterances per dialogue context

**Table 3: Evaluation results on the test sets of the Persona-Chat data and the CMUDoG data. Numbers in bold mean that improvement over the best baseline is statistically significant (t-test, $p$-value $< 0.01$).**

| Metrics / Models | Original Persona | | | Revised Persona | | | CMUDoG | | |
|---|---|---|---|---|---|---|---|---|---|
| | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 |
| IR Baseline [43] | 41.0 | - | - | 20.7 | - | - | - | - | - |
| Starspace [35] | 49.1 | 60.2 | 76.5 | 32.2 | 48.3 | 66.7 | 50.7 | 64.5 | 80.3 |
| Profile Memory [43] | 50.9 | 60.7 | 75.7 | 35.4 | 48.3 | 67.5 | 51.6 | 65.8 | 81.4 |
| KV Profile Memory [43] | 51.1 | 61.8 | 77.4 | 35.1 | 45.7 | 66.3 | 56.1 | 69.9 | 82.4 |
| Transformer [19] | 54.2 | 68.3 | 83.8 | 42.1 | 56.5 | 75.0 | 60.3 | 74.4 | 87.4 |
| DGMN [46] | 67.6 | 80.2 | 92.9 | 58.8 | 62.5 | 87.7 | 65.6 | 78.3 | 91.2 |
| DIM [7] | 78.8 | - | - | 70.7 | - | - | 78.58 | 88.41 | 96.47 |
| RSM-DCK | **79.65** | **90.21** | **97.47** | **71.85** | **84.94** | **95.50** | **79.25** | **88.84** | **96.66** |

**Table 4: Ablation study on the Persona-Chat and CMUDoG dataset.**

| Metrics / Models | Original Persona | | | Revised Persona | | | CMUDoG | | |
|---|---|---|---|---|---|---|---|---|---|
| | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 | $R_{20}$@1 | $R_{20}$@2 | $R_{20}$@5 |
| RSM-DCK | 79.65 | 90.21 | 97.47 | 71.85 | 84.94 | 95.50 | 79.25 | 88.84 | 96.66 |
| RSM-DCK (w/o context selector) | 78.79 | 89.60 | 97.16 | 71.13 | 84.38 | 95.15 | 77.93 | 88.17 | 96.53 |
| RSM-DCK (w/o knowledge selector) | 78.94 | 89.19 | 97.00 | 71.09 | 84.25 | 95.49 | 78.18 | 88.55 | 96.61 |
| RSM-DCK (w/o knowledge post-selection) | 78.87 | 89.36 | 97.27 | 70.79 | 84.58 | 95.43 | 79.19 | 89.09 | 96.82 |
| RSM-DCK (w/o context) | 48.38 | 58.21 | 72.95 | 30.10 | 41.12 | 60.90 | 57.01 | 72.13 | 87.99 |
| RSM-DCK (w/o knowledge) | 62.94 | 77.37 | 91.57 | 63.03 | 77.64 | 91.00 | 74.21 | 86.02 | 95.45 |

is set as 8, and 20 for sentences per document; $l_{p_i} = l_{q_i} = l_r = 40$. The Adam method [12] is applied with learning rate set as 0.00025 for the Persona-Chat dataset with a batch size of 12 and 0.0001 for the CMUDoG dataset with a batch size of 6.

We implement our model with PyTorch and train the model with at most 20 epochs on a 2080ti machine. The early stop is adopted to avoid overfitting. We copy the most results from the DGMN [46]. For the DIM [7] on CMUDoG dataset, we use the codes published at https://github.com/JasonForJoy/DIM to get the results.

## 5.4 Evaluation Results

Table 3 shows the evaluation results on the benchmark datasets. We can find that our proposed model outperforms the baseline models in terms of all metrics. Compared with DIM, the previous best performing baseline on all the two benchmarks, our model achieves 0.9% absolute improvement on $R_{20}$@1 on the Original Persona-Chat dataset, 1.2% absolute improvement on $R_{20}$@1 on the Revised Persona-Chat dataset, and 0.7% absolute improvement on $R_{20}$@1 on the CMUDoG dataset. It is worth noting that the improvement on the CMUDoG dataset is relatively smaller than Persona-Chat. The reason might be that the knowledge is more important for Persona-Chat compared with CMUDoG and the model can benefit more from the selection of the knowledge on the Persona-Chat dataset.

# 6 DISCUSSIONS

In this section, we investigate the effects of the variance of the dialogue context and knowledge collection on the performance of RSM-DCK. First, we conduct an ablation study to demonstrate the effectiveness of RSM-DCK empirically. Second, we explore how the performance of RSM-DCK varies with respect to the number of utterances in the dialogue context. In addition, we visualize the weights of the context and knowledge collection given by the selection mechanism in RSM-DCK through a case study.

## 6.1 Ablations

We perform a series of ablation experiments to investigate the relative importance of the selection mechanism for the context and knowledge. Also, we investigate the relative importance of the context and knowledge collection individually. First, we use the complete model as the baseline, then we conduct ablation experiments as follows:

- **w/o context selector**: we remove the dialogue context selector in the selection layer.
- **w/o knowledge selector**: we remove the document selector in the selection layer.
- **w/o knowledge post-selection**: we remove the post-selection for knowledge in the aggregation layer, and replaced it with a simple mean pooling operation.
- **w/o context**: we remove the dialogue context from the model. The setting means that the response is selected according to the knowledge collection only.
- **w/o knowledge**: we remove the knowledge collection from the model. The model becomes a traditional context-response matching architecture.

Table 4 shows the evaluation results of the ablation studies. We can observe that removing any of the context selector, knowledge selector, and knowledge post-selection leads to the performance drop compared with the complete model, which demonstrates the effectiveness and necessity of each component of RSM-DCK. Besides, we find that the context selector and knowledge selector show a comparable role on the Persona-Chat dataset, while the context

**Table 5: Performance of models across different numbers of utterances in the context.**

| | Original Persona | | | | Revised Persona | | | | CMUDoG | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Utterances number | (0,3] | (4,7] | (8,11] | (12,15] | (0,3] | (4,7] | (8,11] | (12,15] | (0,2] | (3,4] | (5,6] | (7,8] |
| Case Number | 1936 | 1936 | 1936 | 1704 | 1936 | 1936 | 1936 | 1704 | 537 | 537 | 537 | 5026 |
| DGMN $R_{20}$@1 | 73.61 | 71.23 | 66.27 | 61.44 | 62.55 | 58.94 | 59.56 | 58.86 | 79.14 | 74.67 | 70.20 | 69.34 |
| DIM $R_{20}$@1 | 82.85 | 81.56 | 76.76 | 71.48 | **74.33** | 70.76 | 69.47 | 69.60 | 88.12 | 83.43 | 75.98 | 77.32 |
| RSM-DCK $R_{20}$@1 | **84.61** | **81.66** | **77.65** | **74.00** | 72.56 | **71.95** | **72.31** | **70.42** | 87.71 | 83.05 | **79.52** | **77.91** |

| $\bar{s}_c$ | Context |
|---|---|
| 0.147 | hello what are doing today? |
| 0.092 | i am good, i just got off work and tired, i have two jobs. |
| 0.186 | i just got done watching a horror movie |
| 0.131 | i rather read, i have read about 20 books this year. |
| 0.137 | wow! i do love a good horror movie. loving this cooler weather |
| 0.178 | but a good movie is always good. |
| 0.130 | yes! my son is in junior high and i just started letting him watch them too |
| True response | i work in the movies as well. |
| False response | that is great ! are you going to college ? |

**Figure 3: Weights of each utterance in the context.**

| $\bar{s}_k$ | Knowledge collection | $\gamma$ |
|---|---|---|
| 0.278 | i read twenty books a year. | 0.0006 |
| 0.206 | i am a stunt double as my second job. | 0.9987 |
| 0.390 | i only eat kosher. | 0.0001 |
| 0.126 | i was raised in a single parent household. | 0.0005 |

**Figure 4: Weights of each entry in the knowledge collection.**

selector is generally more useful than the knowledge selector on the CMUDoG dataset as the performance of the model drops more when the context selector is removed. According to the results of the last two rows in Table 4, we can observe that removing the dialogue context leads to more degradation of performance of models than removing the knowledge collection, indicating that the dialogue context is more important than the knowledge collection for response selection. It should be noticed that the performance of RSM-DCK on the Persona-Chat dataset drops more dramatically than that on the CMUDoG dataset when the knowledge collection is removed from the matching model. The result implies that knowledge collection is more important for recognizing the proper response candidate on the Persona-Chat dataset.

## 6.2 Length Analysis

We further study how the number of utterances in the dialogue context influences the performance of RSM-DCK by binning the test samples into different buckets according to the number of utterances in the dialogue context. To be noticed, the maximum number of utterances per dialogue context is set as 8 in the CMUDoG dataset, which is different from 4 and 15 set by DGMN and DIM respectively. Thus, we run the codes of DGMN and DIM under our settings. As shown in Table 5, the performance degrades as the number of utterances rises up. The reason is that the topic shifting exists in the dialogue context, and the model may suffer from the new topic unrelated with the dialogue history as the number of utterances rises up, thus leading to performance decrease. Besides, we can also observe that RSM-DCK shows a stronger capability of handling the dialogue context with more utterances than other models. The main reason is that RSM-DCK can select the relevant parts of the dialogue context and thus achieve better understanding of the dialogue context, with the help of the context selector in the selection layer.

## 6.3 Case Study

To further understand how RSM-DCK performs content selection for the dialogue context and knowledge collection, we visualize the relevance score (attention weight) for each entry of the knowledge or the context. Figure 3 shows the relevance scores of each utterance (a.k.a. $\bar{s}_c$) in the dialogue context. Figure 4 shows the pre-selection scores (a.k.a. $\bar{s}_k$) and the post-selection scores (a.k.a. $\gamma$) for each entry in the knowledge collection at the left side and right side respectively. As shown in Figure 3, the latter parts of the dialogue context, with the key word “movie" and key phrase “horror movie", indicate that the topic of the dialogue might be related to movies. Therefore, RSM-DCK rates highly for the third and sixth utterances which are strongly relevant to movies. The result shows that RSM-DCK can properly detect the relevant parts of the dialogue context. As shown in Figure 4, the second sentence in the knowledge collection should be utilized for choosing the true response. Though the pre-selection for the knowledge collection fails, the post-selection still gives the second sentence the highest score, which helps the model out in the dilemma of deciding which response candidate is more sensible.

# 7 CONCLUSION AND FUTURE WORK

In this paper, to overcome the performance drop caused by irrelevant parts contents existing in the dialogue context and document, we propose a multi-turn **R**esponse **S**election **M**odel with carefully designed selection mechanism that can **D**etect the relevant parts of the **C**ontext and **K**nowledge collection (**RSM-DCK**). We conduct experiments on two benchmark datasets and our model achieves new state-of-the-art performance, which shows the superiority of our model. In the future, we would like to apply the content selection mechanism to KG-grounded dialogue systems to select the relevant triples in the knowledge graph and generation-based conversation. Furthermore, we also plan to combine our model with the pre-trained model (such as Bert [3]) to further improve the performance of response selection.

# ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments and valuable suggestions. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC Nos. 61672058 and 61876196). Rui Yan is partially supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI).

# REFERENCES


[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *ACL*. 1657–1668.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*. 4171–4186.

[4] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of Wikipedia: Knowledge-Powered Conversational agents. *arXiv preprint arXiv:1811.01241* (2018).

[5] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

[6] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *CIKM*. 2321–2324.

[7] Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. 2019. Dually Interactive Matching Network for Personalized Response Selection in Retrieval-Based Chatbots. In *EMNLP*. 1845–1854.

[8] Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training Neural Response Selection for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5392–5404.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. 2042–2050.

[11] Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *ICLR*.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*, Yoshua Bengio and Yann LeCun (Eds.).

[13] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al. 2017. AliMe Assist: An Intelligent Assistant for Creating an Innovative E-commerce Experience. In *CIKM*. 2495–2498.

[14] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A Diversity-Promoting Objective Function for Neural Conversation Models. *NAACL* (2015), 110–119.

[15] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A Persona-Based Neural Conversation Model. In *ACL*. 994–1003.

[16] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. *CoRR* abs/1902.04911 (2019).

[17] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. Knowledge Aware Conversation Generation with Explainable Reasoning over Augmented Graphs. In *EMNLP*. 1782–1792.

[18] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL*. 285–294.

[19] Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *EMNLP*. 2775–2779.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

[21] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural Language Inference by Tree-Based Convolution and Heuristic Matching. In *ACL*. 130–136.

[22] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*. 807–814.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[24] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning Personality/Profile to a Chatting Machine for Coherent Conversation Generation. In *IJCAI*. 4279–4285.

[25] Sudha Rao and Hal Daumé III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *ACL*. 2737–2746.

[26] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI*. 3776–3784.

[27] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL*. 1577–1586.

[28] Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From Eliza to XiaoIce: Challenges and Opportunities with Social Chatbots. *Frontiers of IT & EE* (2018).

[29] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-Representation Fusion Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *WSDM*. 267–275.

[30] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues. In *ACL*. 1–11.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[32] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A Dataset for Research on Short-Text Conversations. In *EMNLP*. 935–945.

[33] Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-Based Deep Matching of Short Texts. In *IJCAI*. 1354–1361.

[34] Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to Ask Questions in Open-domain Conversational Systems with Typed Decoders. In *ACL*. 2193–2203.

[35] Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2018. Starspace: Embed all the things!. In *AAAI*.

[36] Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. 2017. A Sequential Matching Framework for Multi-turn Response Selection in Retrieval-based Chatbots. *arXiv preprint arXiv:1710.11344* (2017).

[37] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL*. 496–505.

[38] Chen Xing, Wei Wu, Yu Wu, Ming Zhou, Yalou Huang, and Wei-Ying Ma. 2017. Hierarchical Recurrent Attention Network for Response Generation. *CoRR* abs/1701.07149 (2017).

[39] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*. 55–64.

[40] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting End-to-End Dialogue Systems With Commonsense Knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 4970–4977.

[41] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *EMNLP*. 111–120.

[42] Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. ReCoSa: Detecting the Relevant Contexts with Self-Attention for Multi-turn Dialogue Generation. In *ACL*. 3721–3730.

[43] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243* (2018).

[44] Weinan Zhang, Yiming Cui, Yifa Wang, Qingfu Zhu, Lingzhi Li, Lianqiang Zhou, and Ting Liu. 2018. Context-Sensitive Generation of Open-Domain Conversational Responses. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2437–2447.

[45] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018. Modeling Multi-turn Conversation with Deep Utterance Aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*. 3740–3752.

[46] Xueliang Zhao, Chongyang Tao, Wei Wu, Can Xu, Dongyan Zhao, and Rui Yan. 2019. A Document-grounded Matching Network for Response Selection in Retrieval-based Chatbots. In *IJCAI-19*. 5443–5449.

[47] Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. Personalized Dialogue Generation with Diversified Traits. *CoRR* abs/1901.09672 (2019).

[48] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2017. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. *arXiv preprint arXiv:1704.01074* (2017).

[49] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention.. In *IJCAI*. 4623–4629.

[50] Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A Dataset for Document Grounded Conversations. In *EMNLP*.

[51] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *EMNLP*. 372–381.

[52] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. In *ACL*. 1118–1127.