Filtering before Iteratively Referring for Knowledge-Grounded Response Selection in Retrieval-Based Chatbots

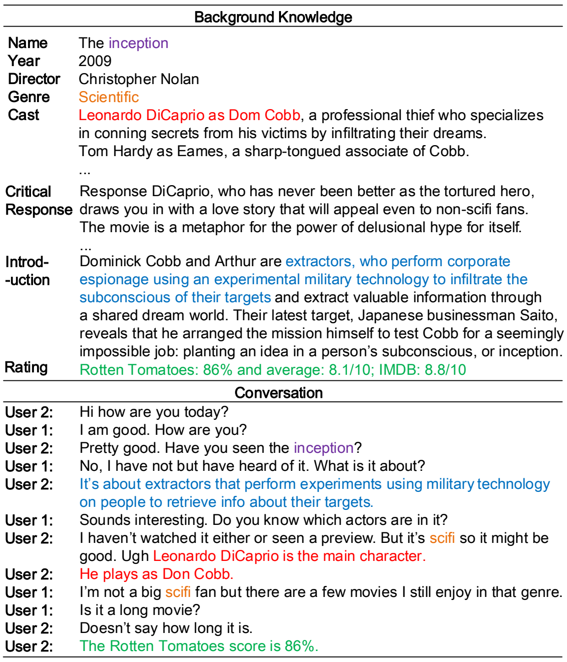
Jia-Chen Gu[1], Zhen-Hua Ling[1][*]; Quan Liu[1,2], Zhigang Chen[2], Xiaodan Zhu[3]
[1]National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Hefei, China
[3]ECE & Ingenuity Labs, Queen's University, Kingston, Canada
gujc@mail.ustc.edu.cn, {zhling, quanliu}@ustc.edu.cn,
zgchen@iflytek.com, xiaodan.zhu@queensu.ca

Abstract

The challenges of building knowledge-grounded retrieval-based chatbots lie in how to ground a conversation on its background knowledge and how to match response candidates with both context and knowledge simultaneously. This paper proposes a method named **F**iltering before **I**teratively **RE**ferring (**FIRE**) for this task. In this method, a context filter and a knowledge filter are first built, which derive knowledge-aware context representations and context-aware knowledge representations respectively by global and bidirectional attention. Besides, the entries irrelevant to the conversation are discarded by the knowledge filter. After that, iteratively referring is performed between context and response representations as well as between knowledge and response representations, in order to collect deep matching features for scoring response candidates. Experimental results show that FIRE outperforms previous methods by margins larger than 2.8% and 4.1% on the PERSONA-CHAT dataset with original and revised personas respectively, and margins larger than 3.1% on the CMU_DoG dataset in terms of top-1 accuracy. We also show that FIRE is more interpretable by visualizing the knowledge grounding process.

1 Introduction

Building a conversational agent with intelligence has received significant attention with the emergence of personal assistants such as Apple Siri, Google Now and Microsoft Cortana. One approach is to building retrieval-based chatbots, which aims to select a potential response from a set of candidates given the conversation context (Lowe et al., 2015; Wu et al., 2017; Zhou et al., 2018b; Gu et al., 2019a; Tao et al., 2019; Gu et al., 2020a).



Background Knowledge	
Name	The inception
Year	2009
Director	Christopher Nolan
Genre	Scientific
Cast	Leonardo DiCaprio as Dom Cobb, a professional thief who specializes in conning secrets from his victims by infiltrating their dreams. Tom Hardy as Eames, a sharp-tongued associate of Cobb. ...
Critical Response	Response DiCaprio, who has never been better as the tortured hero, draws you in with a love story that will appeal even to non-scifi fans. The movie is a metaphor for the power of delusional hype for itself. ...
Introd- -uction	Dominick Cobb and Arthur are extractors, who perform corporate espionage using an experimental military technology to infiltrate the subconscious of their targets and extract valuable information through a shared dream world. Their latest target, Japanese businessman Saito, reveals that he arranged the mission himself to test Cobb for a seemingly impossible job: planting an idea in a person's subconscious, or inception.
Rating	Rotten Tomatoes: 86% and average: 8.1/10; IMDB: 8.8/10

Conversation	
User 2:	Hi how are you today?
User 1:	I am good. How are you?
User 2:	Pretty good. Have you seen the inception?
User 1:	No, I have not but have heard of it. What is it about?
User 2:	It's about extractors that perform experiments using military technology on people to retrieve info about their targets.
User 1:	Sounds interesting. Do you know which actors are in it?
User 2:	I haven't watched it either or seen a preview. But it's scifi so it might be good. Ugh Leonardo DiCaprio is the main character.
User 2:	He plays as Don Cobb.
User 1:	I'm not a big scifi fan but there are a few movies I still enjoy in that genre.
User 1:	Is it a long movie?
User 2:	Doesn't say how long it is.
User 2:	The Rotten Tomatoes score is 86%.

Figure 1: An example from CMU_DoG dataset (Zhou et al., 2018a). Words in the same color are related.

However, real human conversations are often grounded on external knowledge. People may associate relevant background knowledge according to current conversation, and then make their replies based on both context and knowledge. Recently, the tasks of knowledge-grounded response selection (Zhang et al., 2018a; Zhou et al., 2018a) have been set up to simulate this scenario. In these tasks, agents should respond according to not only the given context but also the relevant knowledge, and the knowledge is usually represented as unstructured entries which are common in practice. An example is shown in Figure 1.

Some methods have been proposed for solving these tasks (Mazaré et al., 2018; Zhao et al., 2019; Gu et al., 2019b). In these methods, the semantic representations of context, knowledge

[*]Corresponding author.

and responses candidates are usually derived by encoding models at first. Then, the matching degree between a response candidate and a {context, knowledge} pair is calculated by neural networks. Although these methods are capable of utilizing external knowledge when selecting responses, they still have several deficiencies. First, most of them encode context and knowledge separately, and neglect to ground the conversation on the knowledge and to comprehend the knowledge based on the conversation. Zhao et al. (2019) proposed to alleviate this issue by fusing the *local* matching information between each {context utterance, knowledge entry} pair into their representations. However, each utterance or entry plays different functions in conversations. As shown by the example in Figure 1, some utterances are closely related with background knowledge while some others are irrelevant to knowledge but play the role of connection, such as the *greetings*. Besides, some entries are redundant and are not mentioned in the conversation at all, such as *Year*, *Director* and *Critical Response*. Such *global* functions of utterances and entries were ignored in all existing methods. Second, the model structures used by previous methods to calculate the matching degree between a response candidate and a {context, knowledge} pair were usually *shallow* ones, which constrained the model from learning *deep* matching relationship between them.

Therefore, this paper proposes a method named **F**iltering before **I**teratively **RE**ferring (**FIRE**) to address these issues. First, this method designs a context filter and a knowledge filter at the encoding stage. Different from Zhao et al. (2019), these filters collect the *global* matching information between all context utterances and all knowledge entries bidirectionally. Specifically, the context filter makes the context refer to the knowledge and derives *knowledge-aware context representations*. On the other hand, the knowledge filter derives *context-aware knowledge representations* utilizing the same global attention mechanism. Considering that the knowledge entries are independent of each other and redundant entries may increase the difficulty of response matching, the knowledge filter discards irrelevant entries, which are determined by calculating the similarity between each entry and the whole context.

Second, this method designs an iteratively referring network for calculating the matching degree between a response candidate and a {context, knowledge} pair. This network follows the dual matching framework (Gu et al., 2019b) in which the response refers to the context and the knowledge simultaneously. Motivated by previous studies on attention-over-attention (AoA) (Cui et al., 2017) and interaction-over-interaction (IoI) (Tao et al., 2019) models, this network performs the referring operation iteratively in order to derive *deep* matching information. Specifically, the outputs of each iteration are utilized as the inputs of the next iteration. Then, the outputs of all iterations are aggregated into a set of matching feature vectors for scoring.

We evaluate our proposed method on the PERSONA-CHAT (Zhang et al., 2018a) and CMU_DoG (Zhou et al., 2018a) datasets. Experimental results show that FIRE outperforms previous methods by margins larger than 2.8% and 4.1% on the PERSONA-CHAT dataset with original and revised personas respectively, and margins larger than 3.1% on the CMU_DoG dataset in terms of top-1 accuracy, achieving a new state-of-the-art performance on both tasks.

In summary, the contributions of this paper are three-fold. (1) A **F**iltering before **I**teratively **RE**ferring (**FIRE**) method is proposed, which employs two filtering structures based on global and cross attentions for representing contexts and knowledge, together with an iteratively referring network for scoring response candidates. (2) Experimental results on two datasets demonstrate that our proposed model outperforms state-of-the-art models on the accuracy of response selection. (3) Empirical analysis further verifies the effectiveness of our proposed method.

2 Related Work

2.1 Response Selection

Response selection is an important problem in building retrieval-based chatbots. Existing work on response selection can be categorized according to processing single-turn dialogues (Wang et al., 2013) or multi-turn ones (Lowe et al., 2015; Wu et al., 2017; Zhang et al., 2018b; Zhou et al., 2018b; Gu et al., 2019a; Tao et al., 2019; Gu et al., 2020a,b). Recent studies focused on multi-turn conversations, a more practical setup for real applications. Wu et al. (2017) proposed the sequential matching network (SMN) which accumulated the utterance-response matching information by a recurrent neural network. Zhou et al. (2018b) proposed the deep attention matching network (DAM) to construct representations at different granularities with stacked self-attention. Gu et al. (2019a) proposed the interactive matching network (IMN) to perform the bidirectional and global interactions between the context and the response. Tao et al. (2019) proposed the interaction over interaction (IoI) model which performed matching by stacking multiple interaction blocks. Gu et al. (2020a) proposed the speaker-aware BERT (SA-BERT) to model the speaker change information in pre-trained language models.

2.2 Knowledge-Grounded Chatbots

Chit-chat models suffer from the lack of explicit long-term memory as they are typically trained to produce an utterance given only a very recent dialogue history. Recently, some studies show that chit-chat models can be more diverse and engaging by conditioning them on the background knowledge. Zhang et al. (2018a) released the PERSONA-CHAT dataset which employs the speakers' profile information as the background knowledge. Zhou et al. (2018a) built the CMU_DoG dataset which adopts the Wikipedia articles about popular movies as the background knowledge. Mazaré et al. (2018) proposed to pre-train a model using a large-scale corpus based on Reddit. Zhao et al. (2019) proposed the document-grounded matching network (DGMN) which fused each context utterance with each knowledge entry for representing them. Gu et al. (2019b) proposed a dually interactive matching network (DIM) which performed the interactive matching between responses and contexts and between responses and knowledge respectively.

The FIRE model proposed in this paper makes two major improvements to the state-of-the-art DIM model (Gu et al., 2019b). First, a context filter and a knowledge filter are built to make the representations of context and knowledge aware of each other. Second, an iteratively referring network is designed to collect deep and comprehensive matching information for scoring responses.

3 Task Definition

Given a dataset $\mathcal{D}$, an example is represented as (c, k, r, y). Specifically, $c = \{u_1, u_2, ..., u_{n_c}\}$ represents a context with $\{u_m\}_{m=1}^{n_c}$ as its utterances and n_c as the utterance number. $k = \{e_1, e_2, ..., e_{n_k}\}$ represents a knowledge description with $\{e_n\}_{n=1}^{n_k}$ as its entries and n_k as the entry number. r represents a response candidate. $y \in \{0, 1\}$ denotes a label. $y = 1$ indicates that r is a proper response for (c, k); otherwise, $y = 0$. Our goal is to learn a matching model $g(c, k, r)$ from $\mathcal{D}$. For any context-knowledge-response triple (c, k, r), $g(c, k, r)$ measures the matching degree between (c, k) and r.

4 FIRE Model

Figure 2 shows the overview architecture of our proposed model. The context utterances, knowledge entries and responses are first encoded by a sentence encoder. Then the context and the knowledge are co-filtered by referring to each other. Next, the response refers to the filtered context and knowledge representations iteratively. The outputs of each iteration are aggregated into a matching feature vector, and are utilized as the inputs of next iteration at the same time. Finally, the matching features of all iterations are accumulated for scoring response candidates. Details are provided in following subsections.

4.1 Word Representation

We follow the settings used in DIM (Gu et al., 2019b), which constructs word representations by combining general pre-trained word embeddings, those estimated on the task-specific training set, as well as character-level embeddings, in order to deal with the out-of-vocabulary issue.

Formally, embeddings of the *m*-th utterance in a context, the *n*-th entry in a knowledge description and a response candidate are denoted as $\mathbf{U}_m = \{\mathbf{u}_{m,i}\}_{i=1}^{l_{u_m}}$, $\mathbf{E}_n = \{\mathbf{e}_{n,j}\}_{j=1}^{l_{e_n}}$ and $\mathbf{R} = \{\mathbf{r}_k\}_{k=1}^{l_r}$ respectively, where l_{u_m}, l_{e_n} and l_r are the numbers of words in $\mathbf{U}_m$, $\mathbf{E}_n$ and $\mathbf{R}$ respectively. Each $\mathbf{u}_{m,i}$, $\mathbf{e}_{n,j}$ or $\mathbf{r}_k$ is an embedding vector.

4.2 Sentence Encoder

Note that the encoder can be any existing encoding model. In this paper, the context utterances, knowledge entries and response candidate are encoded by bidirectional long short-term memories (BiLSTMs) (Hochreiter and Schmidhuber, 1997). Detailed calculations are omitted due to limited space. After that, we can obtain the encoded representations for utterances, entries and response, denoted as $\bar{\mathbf{U}}_m = \{\bar{\mathbf{u}}_{m,i}\}_{i=1}^{l_{u_m}}$, $\bar{\mathbf{E}}_n = \{\bar{\mathbf{e}}_{n,j}\}_{j=1}^{l_{e_n}}$ and $\bar{\mathbf{R}} = \{\bar{\mathbf{r}}_k\}_{j=1}^{l_r}$ respectively. Each $\bar{\mathbf{u}}_{m,i}$, $\bar{\mathbf{e}}_{n,j}$ or

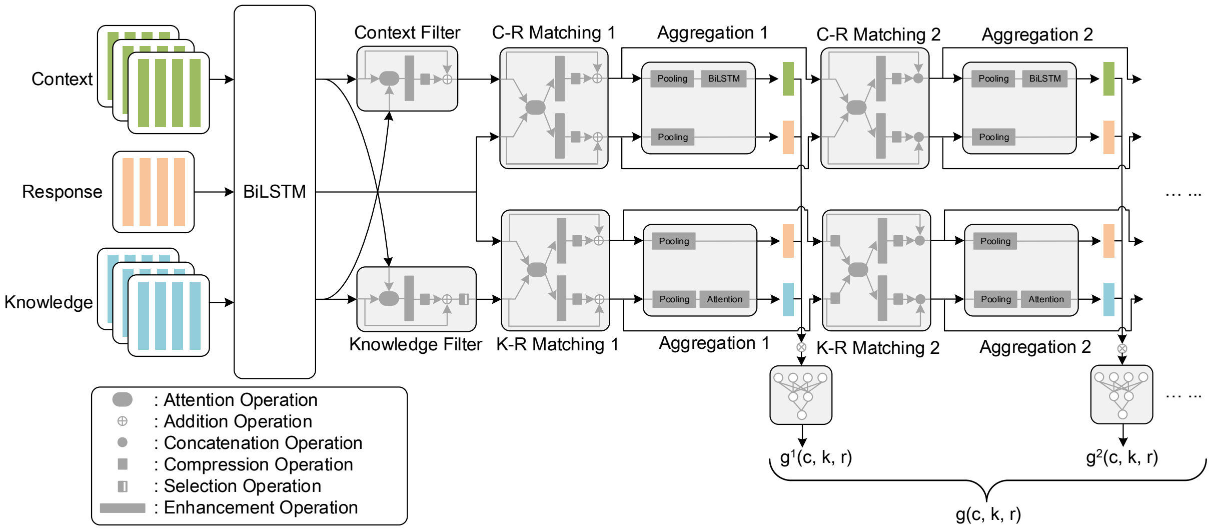


Figure 2: The overview architecture of our proposed FIRE model.

$\bar{\mathbf{r}}_k$ is an embedding vector of d-dimensions. The parameters of these three BiLSTMs are shared in our implementation.

4.3 Context and Knowledge Filters

As illustrated in Figure 1, not every context utterance refers to the knowledge, and not every knowledge entry is mentioned in the conversation. In order to ground the conversation on the knowledge and to comprehend the knowledge based on the conversation, we build a context filter and a knowledge filter in the FIRE model. These two filters obtain *knowledge-aware context representation* $\bar{\mathbf{C}}^0$ and *context-aware knowledge representation* $\bar{\mathbf{K}}^0$, which are further utilized to match with the response.

Context Filter This filter first determines the knowledge that each context token refers to by a *global* attention between the whole context and all knowledge entries. Then, it enhances the representation of each context token with the representations of its relevant knowledge.

Given the set of utterance representations $\{\bar{\mathbf{U}}_m\}_{m=1}^{n_c}$ encoded by the sentence encoder, we concatenate them to form the context representation $\bar{\mathbf{C}} = \{\bar{\mathbf{c}}_i\}_{i=1}^{l_c}$ with $l_c = \sum_{m=1}^{n_c} l_{u_m}$. Also, the knowledge representation $\bar{\mathbf{K}} = \{\bar{\mathbf{k}}_j\}_{j=1}^{l_k}$ with $l_k = \sum_{n=1}^{n_k} l_{e_n}$ is formed similarly by concatenating $\{\bar{\mathbf{E}}_n\}_{n=1}^{n_k}$. Then, a soft alignment is performed by computing the attention weight between each tuple $\{\bar{\mathbf{c}}_i, \bar{\mathbf{k}}_j\}$ as

$$e_{ij} = \bar{\mathbf{c}}_i^\top \cdot \bar{\mathbf{k}}_j. \qquad (1)$$

After that, the global relevance between the context and the knowledge can be obtained using these attention weights. For a word in the context, its relevant representation carried by the knowledge is identified and composed using e_{ij} as

$$\tilde{\mathbf{c}}_i = \sum_{j=1}^{l_k} \frac{exp(e_{ij})}{\sum_{z=1}^{l_k} exp(e_{iz})} \bar{\mathbf{k}}_j, i \in \{1, ..., l_c\}, \qquad (2)$$

where the contents in $\{\bar{\mathbf{k}}_j\}_{j=1}^{l_k}$ that are relevant to $\bar{\mathbf{c}}_i$ are selected to form $\tilde{\mathbf{c}}_i$, and we define $\tilde{\mathbf{C}} = \{\tilde{\mathbf{c}}_i\}_{i=1}^{l_c}$.

To enhance the context representation $\bar{\mathbf{C}}$ with the relevance representation $\tilde{\mathbf{C}}$, the element-wise difference and multiplication between $\{\bar{\mathbf{C}}, \tilde{\mathbf{C}}\}$ are computed, and are then concatenated with the original vectors. This enhancement operation can be written as

$$\hat{\mathbf{C}} = [\bar{\mathbf{C}}; \tilde{\mathbf{C}}; \bar{\mathbf{C}} - \tilde{\mathbf{C}}; \bar{\mathbf{C}} \odot \tilde{\mathbf{C}}], \qquad (3)$$

where $\hat{\mathbf{C}} = \{\hat{\mathbf{c}}_i\}_{i=1}^{l_c}$ and $\hat{\mathbf{c}}_i \in \mathbb{R}^{4d}$. Finally, we compress $\hat{\mathbf{C}}$ and obtain the knowledge-aware context representation $\bar{\mathbf{C}}^0$ as

$$\bar{\mathbf{c}}_i^0 = \mathbf{ReLU}(\hat{\mathbf{c}}_i \cdot \mathbf{W}_c + \mathbf{b}_c) + \bar{\mathbf{c}}_i, \qquad (4)$$

where $\bar{\mathbf{C}}^0 = \{\bar{\mathbf{c}}_i^0\}_{i=1}^{l_c}$, $\mathbf{W}_c \in \mathbb{R}^{4d \times d}$ and $\mathbf{b}_c \in \mathbb{R}^d$ are parameters updated during training.

Here, we define a referring function to summarize above operations in the context filter as

$$\bar{\mathbf{C}}^0 = \mathbf{REFER}(\bar{\mathbf{C}}, \bar{\mathbf{K}}), \qquad (5)$$

where $\bar{\mathbf{C}}$ acts as the *query*, and $\bar{\mathbf{K}}$ acts as the *key* and *value* of the referring function respectively.

Knowledge Filter Similarly, this filter enhances the representation of each knowledge token with the representations of its relevant context. Different from the context filter, an additional selection operation is conducted to directly filter out the knowledge entries with low relevance with the context since the entries are independent of each other.

First, the referring function introduced above is also performed as follows,

$$\bar{\mathbf{K}}^{0'} = \textbf{REFER}(\bar{\mathbf{K}}, \bar{\mathbf{C}}). \tag{6}$$

where $\bar{\mathbf{K}}^{0'}$ is the context-aware knowledge representation and $\bar{\mathbf{K}}^{0'} = \{\bar{\mathbf{E}}_n^{0'}\}_{n=1}^{n_k}$.

Furthermore, the relevance between each entry and the whole conversation is computed in order to determine whether to filter out this entry. We first perform the last-hidden-state pooling over the representations of utterances and entries given by the sentence encoder in Section 4.2. Then, the utterance embedding $\{\bar{\mathbf{u}}_m\}_{m=1}^{n_c}$ and the entry embedding $\{\bar{\mathbf{e}}_n\}_{n=1}^{n_k}$ are obtained. Next, we compute the relevance score for each utterance-entry pair as follows,

$$s_{mn} = \bar{\mathbf{u}}_m^\top \cdot \mathbf{M} \cdot \bar{\mathbf{e}}_n, \tag{7}$$

where $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a matrix that needs to be estimated.

In order to obtain the overall relevance score between each entry and the whole conversation, an aggregation operation is required. Here, we make an assumption that one entry is mentioned only once in the conversation. Thus, for a given entry, its relevance score with the conversation is defined as the maximum relevance score between it and all utterances. Mathematically, we have

$$s_n = \max_m \; s_{mn}. \tag{8}$$

Those entries whose scores are below a threshold γ are considered as uninformative ones for the conversation and are directly filtered out before matching with responses. Mathematically, we have

$$\bar{\mathbf{E}}_n^0 = \max(0, sgn(\sigma(s_n) - \gamma)) \cdot \bar{\mathbf{E}}_n^{0'}, \tag{9}$$

where σ is the sigmoid function and sgn is the sign function. The final filtered knowledge representation is defined as $\bar{\mathbf{K}}^0 = \{\bar{\mathbf{E}}_n^0\}_{n=1}^{n_k}$.

4.4 Iteratively Referring

Zhao et al. (2019) and Gu et al. (2019b) showed that the referring operation between contexts and responses and that between knowledge and responses can both provide useful matching information for response selection. However, the matching information collected by these methods were very *shallow* and *limited*, as each response candidate referred to the context or the knowledge only once in their models. In this paper, we design an iteratively referring network which makes the response refer to the filtered context and knowledge iteratively. Each iteration is capable of capturing additional matching information based on previous ones. Accumulating these iterations can help to derive the *deep* and *comprehensive* matching features for response selection.

Take the context-response matching as an example. The matching strategy adopted here considers the global and bidirectional matching between two sequences. Let $\bar{\mathbf{C}}^l = \{\bar{\mathbf{c}}_i^l\}_{i=1}^{l_c}$ and $\bar{\mathbf{R}}^l = \{\bar{\mathbf{r}}_k^l\}_{k=1}^{l_r}$ be the outputs of the *l*-th iteration, i.e., the inputs of the (*l*+*1*)-th iteration, where $l \in \{0, 1, ..., L-1\}$ and L is the number of iterations. For response representations, we have $\bar{\mathbf{R}}^0 = \bar{\mathbf{R}}$.

First, the context refers to the response by performing the referring function and the *response-aware context representation* $\bar{\mathbf{C}}^{l+1}$ is obtained as

$$\bar{\mathbf{C}}^{l+1} = \textbf{REFER}(\bar{\mathbf{C}}^l, \bar{\mathbf{R}}^l). \tag{10}$$

Bidirectionally, the response refers to the context and the *context-aware response representation* $\bar{\mathbf{R}}^{l+1}$ is obtained as

$$\bar{\mathbf{R}}^{l+1} = \textbf{REFER}(\bar{\mathbf{R}}^l, \bar{\mathbf{C}}^l). \tag{11}$$

$\bar{\mathbf{C}}^{l+1}$ and $\bar{\mathbf{R}}^{l+1}$ are utilized as the input of next iteration. Finally, $\{\bar{\mathbf{C}}^l\}_{l=1}^L$ and $\{\bar{\mathbf{R}}^l\}_{l=1}^L$ are obtained after *L* iterations.

On the other hand, the knowledge-response matching is conducted identically to the context-response matching process introduced above. The *response-aware knowledge representation* $\bar{\mathbf{K}}^l$ and *knowledge-aware response representation* $\bar{\mathbf{R}}^{l*}$ are iteratively updated as

$$\bar{\mathbf{K}}^{l+1} = \textbf{REFER}(\bar{\mathbf{K}}^l, \bar{\mathbf{R}}^{l*}), \tag{12}$$

$$\bar{\mathbf{R}}^{l+1*} = \textbf{REFER}(\bar{\mathbf{R}}^{l*}, \bar{\mathbf{K}}^l), \tag{13}$$

where $\bar{\mathbf{R}}^{0*} = \bar{\mathbf{R}}$. Similarly, we obtain $\{\bar{\mathbf{K}}^l\}_{l=1}^L$ and $\{\bar{\mathbf{R}}^{l*}\}_{l=1}^L$ after *L* iterations.

4.5 Aggregation

These sets of matching matrices $\{\bar{\mathbf{C}}^l\}_{l=1}^L$, $\{\bar{\mathbf{R}}^l\}_{l=1}^L$, $\{\bar{\mathbf{K}}^l\}_{l=1}^L$, and $\{\bar{\mathbf{R}}^{l*}\}_{l=1}^L$ are aggregated into a set of matching feature vectors finally. As shown in Figure 1, we perform the same aggregation operation after each referring iteration. The aggregation strategy in DIM (Gu et al., 2019b) is adopted here.

Let us take the l-th aggregation as an example. First, $\bar{\mathbf{C}}^l$ and $\bar{\mathbf{K}}^l$ are converted back to the matching matrices $\{\bar{\mathbf{U}}_m^l\}_{m=1}^{n_c}$ and $\{\bar{\mathbf{E}}_n^l\}_{n=1}^{n_k}$ for separate utterances and entries. Then, each matching matrix $\bar{\mathbf{U}}_m^l$, $\bar{\mathbf{R}}^l$, $\bar{\mathbf{E}}_n^l$, and $\bar{\mathbf{R}}^{l*}$ are aggregated by max pooling and mean pooling operations to derive their embedding vectors $\bar{\mathbf{u}}_m^l$, $\bar{\mathbf{r}}^l$, $\bar{\mathbf{e}}_n^l$ and $\bar{\mathbf{r}}^{l*}$ respectively. Next, the sequences of $\{\bar{\mathbf{u}}_m^l\}_{m=1}^{n_c}$ and $\{\bar{\mathbf{e}}_n^l\}_{n=1}^{n_k}$ are further aggregated to get the embedding vectors for the context and the knowledge respectively.

As the utterances in a context are chronologically ordered, the utterance embeddings $\{\bar{\mathbf{u}}_m^l\}_{m=1}^{n_c}$ are sent into another BiLSTM following the chronological order of utterances in the context. Combined max pooling and last-hidden-state pooling operations are then performed to derive the context embeddings $\bar{\mathbf{c}}^l$. On the other hand, as the knowledge entries are independent of each other, an attention-based aggregation is designed to derive the knowledge embeddings $\bar{\mathbf{k}}^l$. Readers can refer to Gu et al. (2019b) for more details.

The matching feature vector of the l-th iteration is the concatenation of context, knowledge and response embeddings as

$$\mathbf{m}^l = [\bar{\mathbf{c}}^l; \bar{\mathbf{r}}^l; \bar{\mathbf{k}}^l; \bar{\mathbf{r}}^{l*}], \tag{14}$$

which combines the outputs of both context-response matching and knowledge-response matching.

Last, we obtain a set of matching feature vectors $\{\mathbf{m}^l\}_{l=1}^L$ for all iterations.

4.6 Prediction

Each matching feature vector $\mathbf{m}^l$ is sent into a multi-layer perceptron (MLP) classifier. Here, the MLP is designed to predict the matching degree $g^l(c, k, r)$ between r and (c, k) at l-th iteration. A softmax output layer is adopted in the MLP to return a probability distribution over all response candidates. The probability distributions calculated from all L matching feature vectors are averaged to derive the final distribution for ranking.

4.7 Model Learning

Inspired by Tao et al. (2019), the model parameters of FIRE are learnt by minimizing the summation of cross-entropy losses of MLP at all iterations. By this means, each matching feature vector can be directly supervised by labels in the training set. Furthermore, inspired by Szegedy et al. (2016), we employ the strategy of label smoothing by assigning a small additional confidence ϵ to all candidates, in order to prevent the model from being over-confident. Let Θ denote the parameters of FIRE. The learning objective $\mathcal{L}(\mathcal{D}, \Theta)$ is formulated as

$$\mathcal{L}(\mathcal{D}, \Theta) = -\sum_{l=1}^{L} \sum_{(c,k,r,y)\in\mathcal{D}} (y+\epsilon) log(g^l(c, k, r)). \tag{15}$$

5 Experiments

5.1 Datasets

We tested our proposed method on the PERSONA-CHAT (Zhang et al., 2018a) and CMU_DoG (Zhou et al., 2018a) datasets which both contain dialogues grounded on background knowledge.

The PERSONA-CHAT dataset consists of 8939 complete dialogues for training, 1000 for validation, and 968 for testing. Response selection is performed at every turn of a complete dialogue, which results in 65719 dialogues for training, 7801 for validation, and 7512 for testing in total. Positive responses are true responses from humans and negative ones are randomly sampled by the dataset publishers. The ratio between positive and negative responses is 1:19 in the training, validation, and testing sets. There are 955 personas for training, 100 for validation, and 100 for testing, each consisting of 3 to 5 profile sentences. To make this task more challenging, a version of revised persona descriptions are provided by rephrasing, generalizing, or specializing the original ones.

The CMU_DoG dataset consists of 2881 complete dialogues for training, 196 for validation, and 537 for testing. Response selection is also performed at every turn of a complete dialogue, which results in 36159 dialogues for training, 2425 for validation, and 6637 for testing in total. Since this dataset did not contain negative examples, we adopted the version shared by Zhao et al. (2019), in which 19 negative candidates were randomly sampled for each utterance from the same set.

Model	PERSONA-CHAT						CMU_DoG		
	Original			Revised					
	R@1	**R**@2	**R**@5	**R**@1	**R**@2	**R**@5	**R**@1	**R**@2	**R**@5
Starspace (Wu et al., 2018)	49.1	60.2	76.5	32.2	48.3	66.7	50.7	64.5	80.3
Profile Memory (Zhang et al., 2018a)	50.9	60.7	75.7	35.4	48.3	67.5	51.6	65.8	81.4
KV Profile Memory (Zhang et al., 2018a)	51.1	61.8	77.4	35.1	45.7	66.3	56.1	69.9	82.4
Transformer (Mazaré et al., 2018)	54.2	68.3	83.8	42.1	56.5	75.0	60.3	74.4	87.4
DGMN (Zhao et al., 2019)	67.6	80.2	92.9	58.8	62.5	87.7	65.6	78.3	91.2
DIM (Gu et al., 2019b)	78.8	89.5	97.0	70.7	84.2	95.0	78.7	89.0	97.1
FIRE (Ours)	**81.6**	**91.2**	**97.8**	**74.8**	**86.9**	**95.9**	**81.8**	**90.8**	**97.4**

Table 1: Performance of FIRE and previous methods on the test sets of PERSONA-CHAT and CMU_DoG datasets. The meanings of “Original”, and “Revised” can be found in Section 5.1.

5.2 Evaluation Metrics

We used the same evaluation metrics as the ones in previous work (Zhang et al., 2018a; Zhao et al., 2019). Each model aimed to select k best-matched response from available candidates for the given context and knowledge. Then, the recall of true positive replies, denoted as **R**@k, are calculated as the measurement.

5.3 Training Details

For training FIRE on both PERSONA-CHAT and CMU_DoG datasets, some common configurations were set as follows. The Adam method (Kingma and Ba, 2015) was employed for optimization. The learning rate was initialized as 0.00025 and was exponentially decayed by 0.96 every 5000 steps. Dropout (Srivastava et al., 2014) with a rate of 0.2 was applied to the word embeddings and all hidden layers. The word representation was the concatenation of a 300-dimensional GloVe embedding (Pennington et al., 2014), a 100-dimensional embedding estimated on the training set using the Word2Vec algorithm (Mikolov et al., 2013), and a 150-dimensional character-level embedding estimated by a CNN network that consists of 50 filters and window sizes were set to $\{3,\ 4,\ 5\}$ respectively. The word embeddings were not updated during training. All hidden states of LSTMs had 200 dimensions. The MLP at the prediction layer had 256 hidden units with ReLU (Nair and Hinton, 2010) activation. ϵ used in label smoothing was set to 0.05. The validation set was used to select the best model for testing.

Some configurations were different according to the characteristics of these two datasets. For the PERSONA-CHAT dataset, the maximum number of characters in a word, that of words in a context utterance, of utterances in a context, of words in a response, of words in a knowledge entry, and of entries in a knowledge description were set as 18, 20, 15, 20, 15, and 5 respectively. For the CMU_DoG dataset, these parameters were set as 18, 40, 15, 40, 40 and 20 respectively. Zero-padding was adopted if the number of utterances in a context and the number of knowledge entries in a knowledge description were less than the maximum. Otherwise, we kept the last context utterances or the last knowledge entries. Batch size was set to 16 for PERSONA-CHAT and 4 for CMU_DoG. The hyper-parameter γ was set to 0.3 for original personas and 0.2 for revised personas on the PERSONA-CHAT dataset, as well as 0.2 on the CMU_DoG dataset, which were tuned on the validation sets as shown in Figure 4. The number of iterations L was set to 3 for original and revised personas on the PERSONA-CHAT dataset, as well as 3 on the CMU_DoG dataset, which were tuned on the validation sets as shown in Figure 5.

All code was implemented in the TensorFlow framework (Abadi et al., 2016) and is published to help replicate our results.[1]

5.4 Experimental Results

Table 1 presents the evaluation results of FIRE and previous methods on the PERSONA-CHAT using original or revised personas and on the CMU_DoG dataset. Because the paper proposing DIM (Gu et al., 2019b) only studied the PERSONA-CHAT dataset, we ran its released code to get the performance of DIM on the CMU_DoG dataset.

From Table 1, we can see that FIRE achieved higher top-1 accuracy **R**@1 than all previous methods on both datasets, achieving a new state-

[1] https://github.com/JasonForJoy/FIRE

Model	PERSONA-CHAT		CMU_DoG
	Original	Revised	
	R@1	**R**@1	**R**@1
FIRE	82.3	75.2	83.4
- Ite. Ref.	81.3	73.8	81.6
- Filters	78.9	71.1	78.8
C-R	65.6	66.2	79.7
C-R → Fusion	67.0	66.4	80.9
Filter → C-R	78.8	70.2	81.4
K-R	51.6	34.3	57.8
K-R → Fusion	54.2	39.4	63.1
Filter → K-R	63.6	51.0	73.5

Table 2: The results of ablation tests on the validation sets. Here, C-R denotes context-response matching and K-R denotes knowledge-response matching. The symbol → indicates the order of operations.

of-the-art performance. On the PERSONA-CHAT dataset, the margins were larger than 2.8% and 4.1% when original and revised personas were used respectively. On the CMU_DoG dataset, the margin was larger than 3.1%.

5.5 Analysis

Ablation tests We conducted ablation tests as follows. First, we removed iteratively referring by setting the number of iterations L to one. Then, we removed the two filters. The results on the validation sets are shown in Table 2. We can see the drop of **R**@1 after each step, which demonstrated the effectiveness of both components in FIRE.

To further verify the effectiveness of the context filter, we built three models as follows: (1) a model that only performed the context-response matching without using any knowledge, i.e., the IMN model in Gu et al. (2019b) where readers can refer to for more details; (2) a model that performed the context-response matching first and then fuse the knowledge, i.e., the IMN$_{utr}$ model in Gu et al. (2019b); and (3) a model that filtered the context first and then performed the context-response matching, i.e., our FIRE model with only the upper branch in Figure 2. The evaluation results of these three models on the validation set are shown in Table 2. Since these three models adopted similar context-response matching strategy, we can see that fusion after matching and filtering before matching can both improve the performance of response selection after introducing knowledge. Furthermore, filtering before matching outperformed fusion after matching by a large margin, which demonstrated the effectiveness of the context filter. On the other hand, we also built similar models to further verify the effectiveness of the knowledge filter. The same comparison results were observed from the last three rows of Table 2, which demonstrated its effectiveness.

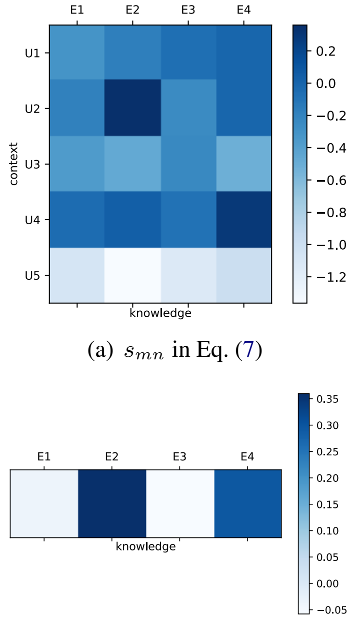
Context Utterances	
U1	hey , are you a student , i traveled a lot , i even studied abroad .
U2	no , i work full time at a nursing home . i am a nurses aide .
U3	nice , i just got a advertising job myself . do you like your job ?
U4	nice . yes i do . caring for people is the joy of my life .
U5	nice my best friend is a nurse , i knew him since kindergarten .
Knowledge Entries	
E1	i have two dogs and one cat .
E2	i work as a nurses aide in a nursing home .
E3	i love to ride my bike .
E4	i love caring for people .

Table 3: Context utterances and knowledge entries of a sample in the test set of the PERSONA-CHAT dataset.

Case Study A case study was conducted to visualize the attention weights in both context and knowledge filters of FIRE model. A sample was used as shown in Table 3. The similarity scores s_{mn} in Eq. (7) for each utterance-entry pair are visualized in Figure 3 (a). The final scores s_n in Eq. (8) for each entry are visualized in Figure 3 (b).

We can see that **U2** and **U4** obtained large attention weights with **E2** and **E4** respectively. Meanwhile, some irrelevant entries **E1** and **E3** obtained small similarity scores with the conversation, which can be filtered out with appropriate threshold. These experimental results verified the effectiveness of the filtering process and the interpretability of the knowledge grounding process.

Knowledge Selection Figure 4 illustrates the validation set performance of FIRE with different threshold γ in the knowledge filter. Here, the number of iterations L was set to 1 for saving computation. When $\gamma = 0$, no knowledge entries were filtered out. From this figure, we can observe a consistent trend that the model performance



(a) s_{mn} in Eq. (7)

(b) s_n in Eq. (8)

Figure 3: Visualizations of (a) s_{mn} in Eq. (7) and (b) s_n in Eq. (8) for a test sample of PERSONA-CHAT. The darker units correspond to larger values.

was improved when increasing γ at the beginning, which indicates that filtering out irrelevant entries indeed helped response selection. Then, the performance started to drop when γ was too large since some indeed relevant entries may be filtered out by mistake.

Iteratively Referring Figure 5 illustrates how the validation set performance of FIRE changed with respect to the number of iterations in iteratively referring. From it, we can see three iterations led to the best performance on both datasets.

Complexity We analysed the time complexity difference between FIRE and DIM. We recorded their inference time over the validation set of PERSONA-CHAT under the configuration of original personas using a GeForce GTX 1080 Ti GPU. It takes FIRE 109.5s and DIM 160.4s to finish the inference, which shows that FIRE is more time-efficient. The reason is that we design a lighter aggregation method in FIRE by replacing recurrent neural network in the aggregation part of DIM with a single-layer non-linear transformation.

6 Conclusion

In this paper, we propose a method named **F**iltering before **I**teratively **RE**ferring (**FIRE**) for utilizing the background knowledge of dialogue agents

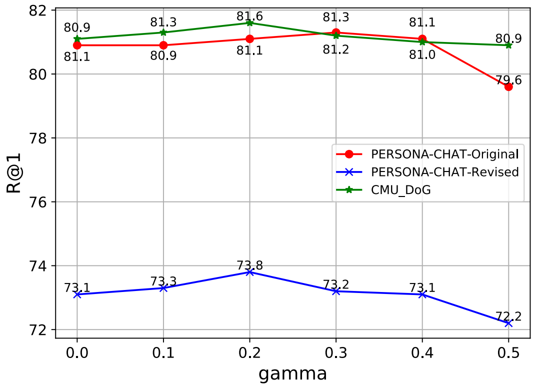


Figure 4: Validation set performance of FIRE with different threshold γ in the knowledge filter.

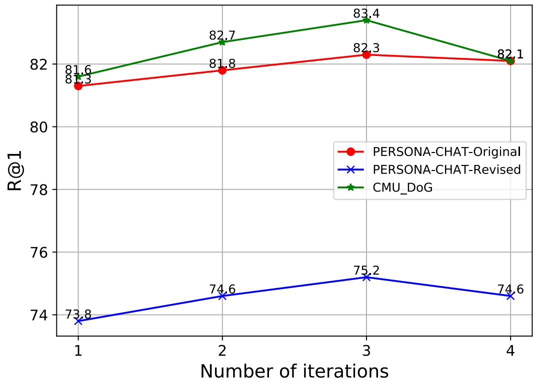


Figure 5: Validation set performance of FIRE with different number of iterations in iteratively referring.

in retrieval-based chatbots. In this method, a context filter and a knowledge filter are first designed to make the representations of context and knowledge aware of each other. Second, an iteratively referring network is built to collect deep and comprehensive matching information for scoring response candidates. Experimental results show that FIRE achieves a new state-of-the-art performance on two datasets. In the future, we will explore better ways of integrating pre-trained language models into our proposed methods for knowledge-grounded response selection.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303001, and in part by the National Nature Science Foundation of China under Grant 61871358 and Grant U1636201. The last author's research is supported by NSERC Discovery Grants and Discovery Accelerator Supplement Grants (DAS).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016.*, pages 265–283.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 593–602. Association for Computational Linguistics.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Wei Si, and Xiaodan Zhu. 2020a. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19-23, 2020.*

Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019a. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2321–2324.

Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2020b. Utterance-to-utterance interactive matching network for multi-turn response selection in retrieval-based chatbots. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:369–379.

Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. 2019b. Dually interactive matching network for personalized response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1845–1854.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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 *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2775–2779.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

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 *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 1–11.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 935–945.

Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2018. Starspace: Embed all the things! In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5569–5577.

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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 496–505.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3740–3752. Association for Computational Linguistics.

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 *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5443–5449.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. 2018a. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 708–713.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018b. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127.