

Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems

Liu Yang¹ Minghui Qiu² Chen Qu¹ Jiafeng Guo³ Yongfeng Zhang⁴ W. Bruce Croft¹

Jun Huang² Haiqing Chen²

¹ Center for Intelligent Information Retrieval, University of Massachusetts Amherst ² Alibaba Group

³ Institute of Computing Technology, Chinese Academy of Sciences ⁴ Dept. of Computer Science, Rutgers University
 {lyang, chenqu, croft}@cs.umass.edu, {minghui.qmh, huangjun.hj, haiqing.chenhq}@alibaba-inc.com
 guojiafeng@ict.ac.cn, yongfeng.zhang@rutgers.edu

ABSTRACT

Intelligent personal assistant systems with either text-based or voice-based conversational interfaces are becoming increasingly popular around the world. Retrieval-based conversation models have the advantages of returning fluent and informative responses. Most existing studies in this area are on open domain “chit-chat” conversations or task / transaction oriented conversations. More research is needed for information-seeking conversations. There is also a lack of modeling external knowledge beyond the dialog utterances among current conversational models. In this paper, we propose a learning framework on the top of deep neural matching networks that leverages external knowledge for response ranking in information-seeking conversation systems. We incorporate external knowledge into deep neural models with pseudo-relevance feedback and QA correspondence knowledge distillation. Extensive experiments with three information-seeking conversation data sets including both open benchmarks and commercial data show that, our methods outperform various baseline methods including several deep text matching models and the state-of-the-art method on response selection in multi-turn conversations. We also perform analysis over different response types, model variations and ranking examples. Our models and research findings provide new insights on how to utilize external knowledge with deep neural models for response selection and have implications for the design of the next generation of information-seeking conversation systems.

ACM Reference Format:

Liu Yang¹ Minghui Qiu² Chen Qu¹ Jiafeng Guo³ Yongfeng Zhang⁴ W. Bruce Croft¹ Jun Huang² Haiqing Chen². 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 8–12, 2018, Ann Arbor, MI, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3209978.3210011>

1 INTRODUCTION

Personal assistant systems, such as Apple Siri, Google Now, Amazon Alexa, and Microsoft Cortana, are becoming ever more widely

used¹. These systems, with either text-based or voice-based conversational interfaces, are capable of voice interaction, information search, question answering and voice control of smart devices. This trend has led to an interest in developing conversational search systems, where users would be able to ask questions to seek information with conversation interactions. Research on speech and text-based conversational search has also recently attracted significant attention in the information retrieval (IR) community.

Existing approaches to building conversational systems include generation-based methods [23, 26] and retrieval-based methods [9, 39–41]. Compared with generation-based methods, retrieval-based methods have the advantages of returning fluent and informative responses. Most work on retrieval-based conversational systems studies response ranking for *single-turn conversation* [34], which only considers a current utterance for selecting responses. Recently, several researchers have been studying *multi-turn conversation* [36, 39, 41, 48], which considers the previous utterances of the current message as the conversation context to select responses by jointly modeling context information, current input utterance and response candidates. However, existing studies are still suffering from the following weaknesses:

(1) **Most existing studies are on open domain chit-chat conversations or task / transaction oriented conversations.** Most current work [9, 23, 26, 39–41] is looking at open domain chit-chat conversations as in microblog data like Twitter and Weibo. There is some research on task oriented conversations [2, 35, 44], where there is a clear goal to be achieved through conversations between the human and the agent. However, the typical applications and data are related to completing transactions like ordering a restaurant or booking a flight ticket. Much less attention has been paid to *information oriented conversations*, which is referred to as *information-seeking conversations* in this paper. Information-seeking conversations, where the agent is trying to satisfy the information needs of the user through conversation interactions, are closely related to conversational search systems. More research is needed on response selection in information-seeking conversation systems.

(2) **Lack of modeling external knowledge beyond the dialog utterances.** Most research on response selection in conversation systems are purely modeling the matching patterns between user input message (either with context or not) and response candidates, which ignores external knowledge beyond the dialog utterances. Similar to Web search, information-seeking conversations

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5657-2/18/07...\$15.00

<https://doi.org/10.1145/3209978.3210011>

¹For example, over 100M installations of Google Now (Google, <http://bit.ly/1wTckVs>); 15M sales of Amazon Echo (GeekWire, <http://bit.ly/2xfZAgX>); more than 141M monthly users of Microsoft Cortana (Windowscentral, <http://bit.ly/2Dv6TVT>).

could be associated with massive external data collections that contain rich knowledge that could be useful for response selection. This is especially critical for information-seeking conversations, since there may be not enough signals in the current dialog context and candidate responses to discriminate a good response from a bad one due to the wide range of topics for user information needs. An obvious research question is how to utilize external knowledge effectively for response ranking. This question has not been well studied, despite the potential benefits for the development of information-seeking conversation systems.

To address these research issues, we propose a learning framework on top of deep neural matching networks that leverages external knowledge for response ranking in information-seeking conversation systems. We study two different methods on integrating external knowledge into deep neural matching networks as follows:

(1) **Incorporating external knowledge via pseudo-relevance feedback.** Pseudo-relevance feedback (PRF) has been proven effective in improving the performance of many retrieval models [3, 12, 17, 25, 46, 47]. The motivation of PRF is to assume a certain number of top-ranked documents from the initial retrieval run to be relevant and use these feedback documents to improve the original query representation. For conversation response ranking, many candidate responses are much shorter compared with conversation context, which could have negative impacts on deep neural matching models. Inspired by the key idea of PRF, we propose using the candidate response as a query to run a retrieval round on a large external collection. Then we extract useful information from the (pseudo) relevant feedback documents to enrich the original candidate response representation.

(2) **Incorporating external knowledge via QA correspondence knowledge distillation.** Previous neural ranking models enhanced the performance of retrieval models such as BM25 and QL, which mainly rely on lexical match information, via modeling semantic match patterns in text [6, 8, 19]. For response ranking in information-seeking conversations, the match patterns between candidate responses and conversation context can be quite different from the well studied lexical and semantic matching. Consider the following sample utterance and response from the conversations in the Microsoft Answers community² shown in Table 1. A Windows user proposed a question about the windows update failure on “restart install”. An expert replied with a response pointing to a potential cause “Norton leftovers”. The match signals between the problem “restart install” and the cause “Norton leftovers” may not be captured by simple lexical and semantic matching. To derive such match patterns, we need to rely on external knowledge to distill QA correspondence information. We propose to extract the “correspondence” regularities between question and answer terms from retrieved external QA pairs. We define this type of match patterns as a “*correspondence match*”, which will be incorporated into deep matching networks as external knowledge to help response selection in information-seeking conversations.

We conduct extensive experiments with three information-seeking conversation data sets: the MSDialog data which contains crawled customer service dialogs from Microsoft Answers community, a popular benchmark data Ubuntu Dialog Corpus (UDC) [16], and another commercial customer service data AliMe from Alibaba

²<https://answers.microsoft.com/>

Table 1: Sample utterance and response from the conversations in the Microsoft Answers community. This figure could be more readable with color print. Note that the purpose of this figure is to illustrate examples and differences among these three types of matches instead of exhaustively labeling all three types of matches between the two texts.

QA Dialog Title: : Windows Update Failure		
Dialog Tags: Windows, Windows 10, Windows update, recovery, backup, PC		
USER: I have Windows10, version 1511, OS Build 10586.1106. For the past year I have tried to upgrade from this without success. Upgrade download OK but on installing only get to 85 - 93% and then on restart install previous version of windows (the 1511 version), I have Windows update assistant installed. Any help or advice on this would be most welcome.		
David		
Responses		
AGENT: James (Microsoft MVP - Windows Client) :		
Response: There's not a doubt in my mind that those Norton " leftovers " is your troublemaker here - but now that the Norton Removal Tool has been deprecated and especially since the new-fangled Norton Remove and Reinstall tool doesn't get rid of the leftovers , a manual upgrade or a clean install of Microsoft Win10 appears to be your only possible resolution here. Feel free to give Norton/Symantec a piece of your mind!		
Term Match: Magenta	Semantic Match: Blue	Correspondence Match: Red

group. We compare our methods with various deep text matching models and the state-of-the-art baseline on response selection in multi-turn conversations. Our methods outperform all baseline methods regrading a variety of metrics.

To sum up, our contributions can be summarized as follows:

(1) **Focusing on information-seeking conversations and building a new benchmark data set.** We target information-seeking conversations to push the boundaries of conversational search models. To this end, we create a new information-seeking conversation data set MSDialog on technical support dialogs of Microsoft products and released it to the research community³.

(2) **Integrating external knowledge into deep neural matching networks for response ranking.** We propose a new response ranking paradigm for multi-turn conversations by incorporating external knowledge into the matching process of dialog context and candidate responses. Under this paradigm, we design two different methods with pseudo relevance feedback and QA correspondence knowledge distillation to integrate external knowledge into deep neural matching networks for response ranking.

(3) **Extensive experimental evaluation on benchmark / commercial data sets and promising results.** Experimental results with three different information-seeking conversation data sets show that our methods outperform various baseline methods including the state-of-the-art method on response selection in multi-turn conversations. We also perform analysis over different response types, model variations and ranking examples to provide insights.

2 RELATED WORK

Our work is related to research on conversational search, neural conversational models and neural ranking models.

Conversational Search. Conversational search has received significant attention with the emerging of conversational devices in the recent years. Radlinski and Craswell described the basic features of conversational search systems [22]. Thomas et al. [29] released the Microsoft Information-Seeking Conversation (MISC) data set, which contains information-seeking conversations with a human intermediary, in a setup designed to mimic software agents such as

³The MSDialog dataset can be downloaded from <https://ciir.cs.umass.edu/downloads/msdialog>. We also released our source code at <https://github.com/yanliuy/NeuralResponseRanking>.

Siri or Cortana. But this data is quite small (in terms of the number of dialogs) for the training of neural models. Based on state-of-the-art advances on machine reading, Kenter and de Rijke [10] adopted a conversational search approach to question answering. Except for conversational search models, researchers have also studied the medium of conversational search. Arguello et al. [1] studied how the medium (e.g., voice interaction) affect user requests in conversational search. Spina et al. studied the ways of presenting search results over speech-only channels to support conversational search [28, 31]. Yang et al. [43] investigated predicting the new question that the user will ask given the past conversational context. Our research targets at the response ranking of information-seeking conversations, with deep matching networks and integration of external knowledge.

Neural Conversational Models. Recent years there are growing interests on research about conversation response generation and ranking with deep learning and reinforcement learning [2, 14, 15, 26, 27, 39–41]. Existing work includes retrieval-based methods [9, 16, 36, 39, 41, 48] and generation-based methods [2, 5, 15, 21, 23, 26, 27, 30, 32]. Sordani et al. [27] proposed a neural network architecture for response generation that is both context-sensitive and data-driven utilizing the Recurrent Neural Network Language Model architecture. Our work is a retrieval-based method. There are some research on multi-turn conversations with retrieval-based method. Wu et al. [36] proposed a sequential matching network that matches a response with each utterance in the context on multiple levels of granularity to distill important matching information. The main difference between our work with their research is that we consider external knowledge beyond dialog context for multi-turn response selection. We show that incorporating external knowledge with pseudo-relevance feedback and QA correspondence knowledge distillation is important and effective for response selection.

Neural Ranking Models. Recently a number of neural ranking models have been proposed for information retrieval, question answering and conversation response ranking. These models could be classified into three categories [6]. The first category is the representation focused models. These models will firstly learn the representations of queries and documents separately and then calculate the similarity score of the learned representations with functions such as cosine, dot, bilinear or tensor layers. A typical example is the DSSM [8] model, which is a feed forward neural network with a word hashing phase as the first layer to predict the click probability given a query string and a document title. The second category is the interaction focused models, which build a query-document term pairwise interaction matrix to capture the exact matching and semantic matching information between the query-document pairs. Then the interaction matrix will be fed into deep neural networks which could be CNN [7, 20, 45], term gating network with histogram or value shared weighting mechanism [6, 42] to generate the final ranking score. In the end, the neural ranking models in the third category combine the ideas of the representation focused models and interaction focused models to joint learn the lexical matching and semantic matching between queries and documents [19, 45]. The deep matching networks used in our research belong to the interaction focused models due to their better performances on a variety of text matching tasks compared with representation focused models [6, 7, 20, 36, 37, 42]. We study different ways to build

Table 2: A summary of key notations in this work. Note that all vectors are denoted with bold cases.

\mathcal{D}	The conversation data set used for training/validation/testing
\mathcal{E}	The collection for the retrieval and distillation of external knowledge
$u_i^t, \mathcal{U}_i, \mathcal{U}$	The t -th utterance of the i -th dialog, all utterances of the i -th dialog and the set of all dialog utterances
$r_i^k, \mathcal{R}_i, \mathcal{R}$	The k -th response candidate for the i -th dialog, all response candidates of the i -th dialog and the set of all candidate responses
$r_i^{k'}$	The k -th expanded response candidate for the i -th dialog
y_i^k, \mathcal{Y}	The label for the k -th response candidate for the i -th dialog and the set of all labels
$f(\cdot)$	The ranking model learnt with \mathcal{D} and \mathcal{E}
$f(\mathcal{U}_i, r_i^k)$	The predicted matching score between \mathcal{U}_i and r_i^k
N	The total number of dialogs in \mathcal{D}
M	The total number of response candidates for \mathcal{U}_i
W	The number of expanded words in response candidates
θ	The language model constructed from the pseudo relevance feedback document set for response candidate expansion
P, \mathcal{P}	The number of top ranked QA posts retrieved from \mathcal{E} and the top ranked QA post set
l_r, l_u	The length of a response candidate and the length of an utterance
d	The number of dimensions of word embedding vectors
$\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$	Interaction matrices between dialog utterance u_i^t and candidate response r_i^k or $r_i^{k'}$ for word embedding similarity, sequence hidden representation similarity and QA correspondence matching similarity
$m_{1,i,j}$	The (i, j) -th element in the interaction matrix \mathbf{M}_1
c	The window size for the utterances in dialog context, which is the maximal number of previous utterances modeled

the interaction matching matrices to capture the matching patterns in term spaces, sequence structures and external knowledge signals between dialog context utterances and response candidates.

3 OUR APPROACH

3.1 Problem Formulation

The research problem of response ranking in information-seeking conversations is defined as follows. We are given an information-seeking conversation data set $\mathcal{D} = \{(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)\}_{i=1}^N$, where $\mathcal{U}_i = \{u_i^1, u_i^2, \dots, u_i^{t-1}, u_i^t\}$ in which $\{u_i^1, u_i^2, \dots, u_i^{t-1}\}$ is the dialog context and u_i^t is the input utterance in the t -th turn. \mathcal{R}_i and \mathcal{Y}_i are a set of response candidates $\{r_i^1, r_i^2, \dots, r_i^M\}_{k=1}^M$ and the corresponding binary labels $\{y_i^1, y_i^2, \dots, y_i^M\}$, where $y_i^k = 1$ denotes r_i^k is a true response for \mathcal{U}_i . Otherwise $y_i^k = 0$. In order to integrate external knowledge, we are also given an external collection \mathcal{E} , which is related to the topics discussed in conversation \mathcal{U} . Our task is to learn a ranking model $f(\cdot)$ with \mathcal{D} and \mathcal{E} . For any given \mathcal{U}_i , the model should be able to generate a ranking list for the candidate responses \mathcal{R}_i with $f(\cdot)$. The external collection \mathcal{E} could be any massive text corpus. In our paper, \mathcal{E} are historical QA posts in Stack Overflow data dump ⁴ for MSDialog, AskUbuntu data dump ⁵ for Ubuntu Dialog Corpus and product QA pairs for AliMe data.

3.2 Method Overview

In the following sections, we describe the proposed learning framework built on the top of deep matching networks and external knowledge for response ranking in information-seeking conversations. A summary of key notations in this work is presented in Table 2. In general, there are three modules in our learning framework:

⁴<https://stackoverflow.com/>

⁵<https://askubuntu.com/>

Table 3: Statistics of external collections for QA pairs retrieval and knowledge extraction. Note that “#QWithAcceptedA” means “number of questions with an accepted answer”. The other names use similar abbreviations.

Collection Name	SOTwoYears	AskUbuntu
StartDate	12/4/2015	7/28/2010
EndDate	9/1/2017	9/1/2017
#QAPosts	9,563,530	629,198
#Time	2 Years	7 years
XMLFileDiskSize	17GB	799MB
#Question	4,188,937	271,233
%QWithAcceptedA	41.82%	34.01%
%QWithAtLeastOneA	75.89%	78.84%

(1) **Information retrieval (IR) module:** Given the information seeking conversation data \mathcal{D} and external QA text collection \mathcal{E} , this module is to retrieve a small relevant set of QA pairs \mathcal{P} from \mathcal{E} with the response candidate \mathcal{R} as the queries. These retrieved QA pairs \mathcal{P} become the source of external knowledge.

(2) **External knowledge extraction (KE) module:** Given the retrieved QA pairs \mathcal{P} from the IR module, this module will extract useful information as term distributions, term co-occurrence matrices or other forms as external knowledge.

(3) **Deep matching network (DMN) module:** This is the module to model the extracted external knowledge from \mathcal{P} , dialog utterances \mathcal{U}_i and the response candidate r_i^k to learn the matching pattern, over which it will accumulate and predict a matching score $f(\mathcal{U}_i, r_i^k)$ for \mathcal{U}_i and r_i^k .

We explore two different implementations under this learning framework as follows: 1) Incorporating external knowledge into deep matching networks via pseudo-relevance feedback (DMN-PRF). The architecture of DMN-PRF model is presented in Figure 1. 2) Incorporating external knowledge via QA correspondence knowledge distillation (DMN-KD). The architecture of DMN-KD model is presented in Figure 2. We will present the details of these two models in Section 3.3 and Section 3.4.

3.3 Deep Matching Networks with Pseudo-Relevance Feedback

3.3.1 Relevant QA Posts Retrieval. We adopt different QA text collections for different conversation data (e.g. Stack Overflow data for MSDialog, AskUbuntu for UDC). The statistics of these external collections are shown in Table 3. We download the data dumps for Stack Overflow and AskUbuntu from archive.org⁶. We index the QA posts in Stack Overflow in most recent two years and all the QA posts in AskUbuntu. Then we use the response candidate r_i^k as the query to retrieve top P ⁷ QA posts with BM25 as the source for external knowledge.

3.3.2 Candidate Response Expansion. The motivation of Pseudo-Relevance Feedback (PRF) is to extract terms from the top-ranked documents in the first retrieval results to help discriminate relevant documents from irrelevant ones [3]. The expansion terms are extracted either according to the term distributions (e.g. extract the most frequent terms) or extracted from the most specific terms (e.g. extract terms with the maximal IDF weights) in feedback

documents. Given the retrieved top QA posts \mathcal{P} from the previous step, we compute a language model $\theta = P(w|\mathcal{P})$ using \mathcal{P} . Then we extract the most frequent W ⁸ terms from θ as expansion terms for response candidate r_i^k and append them at the end of r_i^k . For the query r_i^k , we perform several preprocessing steps including tokenization, punctuation removal and stop words removal. QA posts in both Stack Overflow and AskUbuntu have two fields: “Body” and “Title”. We choose to search the “Body” field since we found it more effective in experiments.

3.3.3 Interaction Matching Matrix. The expanded response candidates and dialog contexts will be modeled by a deep neural matching network. Given an expanded response $r_i^{k'}$ and an utterance u_i^t in the context \mathcal{U}_i , the model firstly looks up a global embedding dictionary to represent $r_i^{k'}$ and u_i^t as two sequences of embedding vectors $E(r_i^{k'}) = [e_{r,1}, e_{r,2}, \dots, e_{r,l_r}]$ and $E(u_i^t) = [e_{u,1}, e_{u,2}, \dots, e_{u,l_u}]$, where $e_{r,i} \in \mathbb{R}^d$, $e_{u,i} \in \mathbb{R}^d$ are the embedding vectors of the i -th word in $r_i^{k'}$ and u_i^t respectively. Given these two word embedding sequences, there are two different methods to learn matching patterns: representation focused methods and interaction focused methods [6]. Here we adopt the interaction focused methods due to their better performances over a number of text matching tasks [7, 20, 33, 42]. Specifically, the model builds two interaction matrices with $E(r_i^{k'}) \in \mathbb{R}^{d \times l_r}$ and $E(u_i^t) \in \mathbb{R}^{d \times l_u}$: a word pairwise similarity matrix \mathbf{M}_1 and a sequence hidden representation similarity matrix \mathbf{M}_2 . \mathbf{M}_1 and \mathbf{M}_2 will be two input channels of a convolutional neural network (CNN) to learn important matching features, which will be aggregated by the final BiGRU layer and a multi-layer perceptron (MLP) to generate a matching score.

Specifically, in the input channel one, $\forall i, j$, the element $m_{1,i,j}$ in the \mathbf{M}_1 is defined by $m_{1,i,j} = e_{r,i}^T \cdot e_{u,j}$. \mathbf{M}_1 models the word pairwise similarity between $r_i^{k'}$ and u_i^t via the dot product similarity between the embedding representations.

For input channel two, we firstly employ bidirectional gated recurrent units (BiGRU) [4] to encode $r_i^{k'}$ and u_i^t into two hidden representations. A BiGRU consists two GRUs that run in opposite directions on sequence $E(r_i^{k'})$: a forward GRUs processing the sequence as it is ordered, and another backward GRUs processing the sequence in its reverse order. These two GRUs will generate two sequences of hidden states $(\vec{\mathbf{h}}_1, \dots, \vec{\mathbf{h}}_{l_r})$ and $(\overleftarrow{\mathbf{h}}_1, \dots, \overleftarrow{\mathbf{h}}_{l_r})$. BiGRU then concatenates the forward and the backward hidden states to form the final hidden vectors for $r_i^{k'}$ as $\mathbf{h}_i = [\vec{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i]_{i=1}^{l_r}$. More specifically, $\forall i$, the hidden state vector $\mathbf{h}_i \in \mathbb{R}^O$ is calculated by the following formulas:

$$\begin{aligned} \mathbf{z}_i &= \sigma(\mathbf{W}_z \mathbf{e}_{r,i} + \mathbf{U}_z \vec{\mathbf{h}}_{i-1} + \mathbf{b}_z) \\ \mathbf{r}_i &= \sigma(\mathbf{W}_r \mathbf{e}_{r,i} + \mathbf{U}_r \vec{\mathbf{h}}_{i-1} + \mathbf{b}_r) \\ \tilde{\mathbf{h}}_i &= \tanh(\mathbf{W}_h \mathbf{e}_{r,i} + \mathbf{U}_h (\mathbf{r}_i \circ \vec{\mathbf{h}}_{i-1}) + \mathbf{b}_h) \\ \overleftarrow{\mathbf{h}}_i &= (1 - \mathbf{z}_i) \circ \vec{\mathbf{h}}_{i-1} + \mathbf{z}_i \circ \tilde{\mathbf{h}}_i \end{aligned} \quad (1)$$

where \mathbf{z}_i and \mathbf{r}_i are an update gate and a reset gate respectively. $\mathbf{e}_{r,i}, \vec{\mathbf{h}}_i$ are the input and hidden state output of the network at time step i . $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h, \mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h$ and $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h$ are parameter matrices and bias vectors to be learned. The backward hidden state $\overleftarrow{\mathbf{h}}_i \in \mathbb{R}^O$ is computed in a similar way according to Equation 1. The

⁶<https://archive.org/download/stackexchange>

⁷In our experiments, we set $P = 10$.

⁸In our experiments, we set $W = 10$.

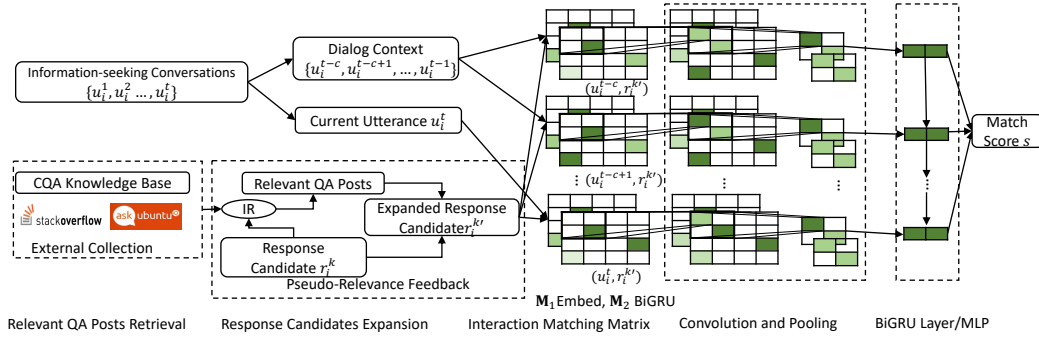


Figure 1: The architecture of DMN-PRF model for conversation response ranking.

hidden vectors for the dialog utterance u_i^t can be obtained in the same procedure. Given the hidden vectors of $r_i^{k'}$ and u_i^t , we calculate element $m_{2,i,j}$ in the sequence hidden representation similarity matrix M_2 by $m_{2,i,j} = \mathbf{h}_{r_i^{k'}}^T \cdot \mathbf{h}_{u_i^t}$. BiGRU models the neighbor context information around words from two directions and encode the text sequences into hidden vectors. Thus M_2 matches $r_i^{k'}$ and u_i^t with local sequence structures such as phrases or text segments.

3.3.4 Convolution and Pooling Layers. The interaction matrices M_1 and M_2 are then fed into a CNN to learn high level matching patterns as features. CNN alternates convolution and max-pooling operations over these input channels. Let $\mathbf{z}^{(l,k)}$ denote the output feature map of the l -th layer and k -th kernel, the model will do convolution operations and max-pooling operations according to the following equations.

Convolution. Let $r_w^{(l,k)} \times r_h^{(l,k)}$ denote the shape of the k -th convolution kernel in the l -th layer, the convolution operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \sigma \left(\sum_{k'=0}^{K_l-1} \sum_{s=0}^{r_w^{(l,k)}-1} \sum_{t=0}^{r_h^{(l,k)}-1} \mathbf{w}_{s,t}^{(l+1,k)} \cdot \mathbf{z}_{i+s,j+t}^{(l,k')} + b^{(l+1,k)} \right) \quad (2)$$

$\forall l = 0, 2, 4, 6, \dots$

where σ is the activation function ReLU, and $\mathbf{w}_{s,t}^{(l+1,k)}$ and $b^{(l+1,k)}$ are the parameters of the k -th kernel on the $(l+1)$ -th layer to be learned. K_l is the number of kernels on the l -th layer.

Max Pooling. Let $p_w^{(l,k)} \times p_h^{(l,k)}$ denote the shape of the k -th pooling kernel in the l -th layer, the max pooling operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \max_{0 \leq s < p_w^{(l,k)}} \max_{0 \leq t < p_h^{(l,k)}} \mathbf{z}_{i+s,j+t}^{(l,k)} \quad \forall l = 1, 3, 5, 7, \dots, \quad (3)$$

3.3.5 BiGRU Layer and MLP. Given the output feature representation vectors learned by CNN for utterance-response pairs ($r_i^{k'}$, u_i^t), we add another BiGRU layer to model the dependency and temporal relationship of utterances in the conversation according to Equation 1 following the previous work [36]. The output hidden states $\mathbf{H}_c = [\mathbf{h}'_1, \dots, \mathbf{h}'_c]$ will be concatenated as a vector and fed into a multi-layer perceptron (MLP) to calculate the final matching score $f(\mathcal{U}_i, r_i^{k'})$ as

$$f(\mathcal{U}_i, r_i^{k'}) = \sigma_2(\mathbf{w}_2^T \cdot \sigma_1(\mathbf{w}_1^T \mathbf{H}_c + \mathbf{b}_1) + \mathbf{b}_2) \quad (4)$$

where $\mathbf{w}_1, \mathbf{w}_2, \mathbf{b}_1, \mathbf{b}_2$ are model parameters. σ_1 and σ_2 are tanh and softmax functions respectively.

3.3.6 Model Training. For model training, we consider a pairwise ranking learning setting. The training data consists of triples $(\mathcal{U}_i, r_i^{k+}, r_i^{k-})$ where r_i^{k+} and r_i^{k-} denote the positive and the negative response candidate for dialog utterances \mathcal{U}_i . Let Θ denote all the parameters of our model. The pairwise ranking-based hinge loss function is defined as:

$$\mathcal{L}(\mathcal{D}, \Theta) = \sum_{i=1}^I \max(0, \epsilon - f(\mathcal{U}_i, r_i^{k+}) + f(\mathcal{U}_i, r_i^{k-})) + \lambda \|\Theta\|_2^2 \quad (5)$$

where I is the total number of triples in the training data \mathcal{D} . $\lambda \|\Theta\|_2^2$ is the regularization term where λ denotes the regularization coefficient. ϵ denotes the margin in the hinge loss. The parameters of the deep matching network are optimized using back-propagation with Adam algorithm [11].

3.4 Deep Matching Networks with QA Correspondence Knowledge Distillation

In addition to the DMN-PRF model presented in Section 3.3, we also propose another model for incorporating external knowledge into conversation response ranking via QA correspondence knowledge distillation, which is referred to as DMN-KD model in this paper. The architecture of DMN-KD model is presented in Figure 2. Compared with DMN-PRF, the main difference is that the CNN of DMN-KD will run on an additional input channel M_3 denoted as blue matrices in Figure 2, which captures the correspondence matching patterns of utterance terms and response terms in relevant external QA pairs retrieved from \mathcal{E} . Specifically, we firstly use the response candidate r_i^k as the query to retrieve a set of relevant QA pairs⁹ \mathcal{P} . Suppose $\mathcal{P} = \{Q, \mathcal{A}\} = \{(Q_1, A_1), (Q_2, A_2), \dots, (Q_P, A_P)\}$, where (Q_p, A_p) denotes the p -th QA pair. Given a response candidate r_i^k and a dialog utterance u_i^t in dialog \mathcal{U}_i , the model will compute the term co-occurrence information as the *Positive Pointwise Mutual Information* (PPMI) of words of r_i^k and u_i^t in retrieved QA pair set $\{Q, \mathcal{A}\}$. Let $[w_{r,1}, w_{r,2}, \dots, w_{r,l_r}]$ and $[w_{u,1}, w_{u,2}, \dots, w_{u,l_u}]$ denote the word sequence in r_i^k and u_i^t . We construct a QA term correspondence matching matrix M_3 as the third input channel of CNN for r_i^k and u_i^t with the PPMI statistics from $\{Q, \mathcal{A}\}$. More specifically, $\forall i, j$, the element $m_{3,i,j}$ in M_3 is computed as

$$\begin{aligned} m_{3,i,j} &= \text{PPMI}(w_{r,i}, w_{u,j} | \{Q, \mathcal{A}\}) \\ &= \max(0, \log \frac{\sum_{p'=1}^P p(w_{r,i} \in A_{p'}, w_{u,j} \in Q_{p'} | Q_{p'}, A_{p'})}{p(w_{r,i} | \mathcal{A}) \cdot p(w_{u,j} | Q)}) \end{aligned} \quad (6)$$

⁹Note that we want QA pairs here instead of question posts or answer posts, since we would like to extract QA term co-occurrence information with these QA pairs.

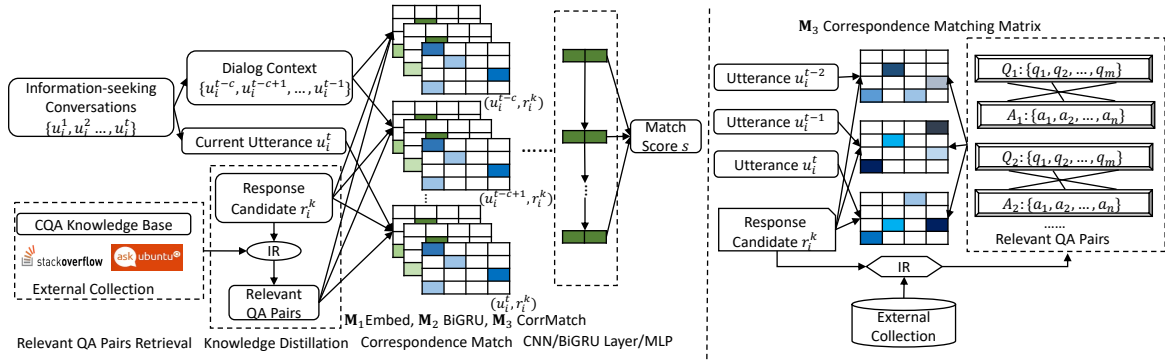


Figure 2: The left figure shows the architecture of DMN-KD model for conversation response ranking. The input channel M_3 denoted as blue matrices capture the correspondence matching patterns of utterance terms and response terms in relevant external QA pairs retrieved from \mathcal{E} . Note that we omit the details for CNN layers here to save spaces as they have been visualized in Figure 1. The right figure shows the detailed pipeline of external relevant QA pairs retrieval and QA correspondence matching knowledge distillation in DMN-KD model.

Table 4: The statistics of experimental datasets, where C denotes context and R denotes response. # Cand. per C denotes the number of candidate responses per context.

Data	UDC			MSDialog			AliMe		
Items	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
# C-R pairs	1000k	500k	500k	173k	37k	35k	51k	6k	6k
# Cand. per C	2	10	10	10	10	10	15	15	15
# + Cand. per C	1	1	1	1	1	1	2.9	2.8	2.9
Min # turns per C	1	2	1	2	2	2	2	2	2
Max # turns per C	19	19	19	11	11	11	3	3	3
Avg # turns per C	10.1	10.1	10.1	5.0	4.9	4.4	2.4	2.1	2.2
Avg # words per C	116	116	116	271	263	227	38	35	34
Avg # words per R	22.1	22.1	22.1	66.7	67.6	66.8	4.9	4.7	4.6

where $w_{r,i}$ and $w_{u,j}$ denote the i -th word in the response candidate and j -th word in the dialog utterance. The intuition is that the PPMI between $w_{r,i}$ and $w_{u,j}$ in the top retrieved relevant QA pair set $\{Q, \mathcal{A}\}$ could encode the correspondence matching patterns between $w_{r,i}$ and $w_{u,j}$ in external relevant QA pairs. Thus M_3 is the extracted QA correspondence knowledge from the external collection \mathcal{E} for r_i^k and u_i^t . These correspondence matching knowledge capture relationships such as “(Problem Descriptions, Solutions)”, “(Symptoms, Causes)”, “(Information Request, Answers)”, etc. in the top ranked relevant QA pair set $\{Q, \mathcal{A}\}$. They will help the model better discriminate a good response candidate from a bad response candidate given the dialog context utterances. To compute the co-occurrence count between $w_{r,i}$ and $w_{u,j}$, we count all word co-occurrences considering A_p and Q_p as bag-of-words as we found this setting is more effective in experiments.

4 EXPERIMENTS

4.1 Data Set Description

We evaluated our method with three data sets: Ubuntu Dialog Corpus (UDC), MSDialog, and AliMe data consisting of a set of customer service conversations in Chinese from Alibaba.

4.1.1 Ubuntu Dialog Corpus. The Ubuntu Dialog Corpus (UDC) [16] contains multi-turn technical support conversation data collected from the chat logs of the Freenode Internet Relay Chat

(IRC) network. We used the data copy shared by Xu et al.[38], in which numbers, urls and paths are replaced by special placeholders. It is also used in several previous related works [36]¹⁰. It consists of 1 million context-response pairs for training, 0.5 million pairs for validation and 0.5 million pairs for testing. The statistics of this data is shown in Table 4. The positive response candidates in this data come from the true responses by human and negative response candidates are randomly sampled.

4.1.2 MSDialog. In addition to UDC, we also crawled another technical support conversation data from the Microsoft Answer community, which is a QA forum on topics about a variety of Microsoft products. We firstly crawled 35,536 dialogs about 76 different categories of Microsoft products including “Windows”, “IE”, “Office”, “Skype”, “Surface”, “Xbox”, etc.¹¹ Then we filtered dialogs whose number of turns are out of the range [3, 99]. After that we split the data into training/validation/testing partitions by time. Specifically, the training data contains 25,019 dialogs from “2005-11-12” to “2017-08-20”. The validation data contains 4,654 dialogs from “2017-08-21” to “2017-09-20”. The testing data contains 5,064 dialogs from “2017-09-21” to “2017-10-04”.

The next step is to generate the dialog context and response candidates. For each dialog, we assigned “User” label to the first participant who proposed the question leading to this information-seeking conversation, and “Agent” label to the other participants who provided responses. The “Agent” in our data could be Microsoft customer service staff, a Microsoft MVP (Most Valuable Professional) or a user from the Microsoft Answer community. Then for each utterance by the “User” u_i^t ¹², we collected the previous c utterances as the dialog context, where $c = \min(t-1, 10)$ and $t-1$ is the total number of utterances before u_i^t . The true response by the “Agent” becomes the positive response candidate. For the negative response candidates, we adopted negative sampling to construct

¹⁰The data can be downloaded from <https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntu%20data.zip?dl=0>

¹¹Note that some categories are more fine-grained, such as “Outlook_Calendar”, “Outlook_Contacts”, “Outlook_Email”, “Outlook_Messaging”, etc.

¹²We consider the utterances by the user except the first utterance, since there is no associated dialog context with it.

them following previous work [16, 33, 36]. For each dialog context, we firstly used the true response as the query to retrieve the top 1,000 results from the whole response set of agents with BM25. Then we randomly sampled 9 responses from them to construct the negative response candidates. The statistics of MSDialog data is presented in Table 4. For data preprocessing, we performed tokenization and punctuation removal. Then we removed stop words and performed word stemming. For neural models, we also removed words that appear less than 5 times in the whole corpus.

4.1.3 AliMe Data. We collected the chat logs between customers and a chatbot AliMe from “2017-10-01” to “2017-10-20” in Alibaba. The chatbot is built based on a question-to-question matching system¹³ [13], where for each query, it finds the most similar candidate question in a QA database and return its answer as the reply. It indexes all the questions in our QA database using Lucence¹⁴. For each given query, it uses TF-IDF ranking algorithm to call back candidates. To form our data set, we concatenated utterances within three turns¹⁵ to form a query, and used the chatbot system to call back top-K¹⁶ most similar candidate questions as candidate “responses”.¹⁷ We then asked a business analyst to annotate the candidate responses, where a “response” is labeled as positive if it matches the query, otherwise negative. In all, we have annotated 63,000 context-response pairs, where we use 51,000 as training, 6,000 for testing, and 6,000 for validation shown in Table 4. Note that we have included human evaluation in AliMe data. Furthermore, if the confidence score of answering a given user query is low, the system will prompt three top related questions for users to choose. We collected such user click logs as our external data, where we treat the clicked question as positive and the others as negative. We collected 510,000 clicked questions with answers from the click logs in total as the source of external knowledge.

4.2 Experimental Setup

4.2.1 Baselines. We consider different types of baselines for comparison, including traditional retrieval models, deep text matching models and the state-of-the-art multi-turn conversation response ranking method as the following:

BM25. This method uses the dialog context as the query to retrieve response candidates for response selection. We consider BM25 model [24] as the retrieval model.

ARC-II. ARC-II is an interaction focused deep text matching architectures proposed by Hu et al. [7], which is built directly on the interaction matrix between the dialog context and response candidates. A CNN is running on the interaction matrix to learn the matching representation score.

MV-LSTM. MV-LSTM [33] is a neural text matching model that matches two sequences with multiple positional representations learned by a Bi-LSTM layer.

DRMM. DRMM [6] is a deep relevance matching model for ad-hoc retrieval. We implemented a variant of DRMM for short text matching. Specifically, the matching histogram is replaced by a

top-k max pooling layer and the remaining part is the same with the original model.

Duet. Duet [19] is the state-of-the-art deep text matching model that jointly learns local lexical matching and global semantic matching between the two text sequences.

SMN. Sequential Matching Network (SMN) [36] is the state-of-the-art deep neural architecture for multi-turn conversation response selection. It matches a response candidate with each utterance in the context on multiple levels of granularity and then adopts a CNN network to distill matching features. We used the TensorFlow¹⁸ implementation of SMN shared by authors [36]¹⁹.

We also consider a degenerated version of our model, denoted as **DMN**, where we do not incorporate external knowledge via pseudo-relevance feedback or QA correspondence knowledge distillation. Finally, we consider a baseline **BM25-PRF**, where we incorporate external knowledge into BM25 by matching conversation context with the expanded responses as in Section 3.3.2 using BM25 model.

4.2.2 Evaluation Methodology. For the evaluation metrics, we adopted mean average precision (MAP), Recall@1, Recall@2, and Recall@5 following previous related works [16, 36]. For UDC and MSDialog, MAP is equivalent to the mean reciprocal rank (MRR) since there is only one positive response candidate per dialog context. For AliMe data, each dialog context could have more than one positive response candidates.

4.2.3 Parameter Settings. All models were implemented with TensorFlow and MatchZoo²⁰ toolkit. Hyper-parameters are tuned with the validation data. For the hyper-parameter settings of DMN-KD and DMN-PRF models, we set the window size of the convolution and pooling kernels as (3, 3). The number of convolution kernels is 8 for UDC and 2 for MSDialog. The dimension of the hidden states of BiGRU layer is set as 200 for UDC and 100 for MSDialog. The dropout rate is set as 0.3 for UDC and 0.6 for MSDialog. All models are trained on a single Nvidia Titan X GPU by stochastic gradient descent with Adam[11] algorithm. The initial learning rate is 0.001. The parameters of Adam, β_1 and β_2 are 0.9 and 0.999 respectively. The batch size is 200 for UDC and 50 for MSDialog. The maximum utterance length is 50 for UDC and 90 for MSDialog. The maximum conversation context length is set as 10 following previous work [36]. We padded zeros if the number of utterances in a context is less than 10. Otherwise the most recent 10 utterances will be kept. For DMN-PRF, we retrieved top 10 QA posts and extracted 10 terms as response expansion terms. For DMN-KD, we retrieved top 10 question posts with accepted answers. For the word embeddings used in our experiments, we trained word embeddings with the Word2Vec tool [18] with the Skip-gram model using our training data. The max skip length between words and the number of negative examples is set as 5 and 10 respectively. The dimension of word vectors is 200. Word embeddings will be initialized by these pre-trained word vectors and updated during the training process.

¹³ Interested readers can access AliMe Assist through the Taobao App, or the web version via <https://consumerservice.taobao.com/online-help>

¹⁴ <https://lucene.apache.org/core/>

¹⁵ The majority (around 85%) of conversations in the data set are within 3 turns.

¹⁶ We set $K=15$.

¹⁷ A “response” here is a question in our system.

¹⁸ <https://www.tensorflow.org/>

¹⁹ The reported SMN results with the code from authors are on the raw data sets of UDC and MSDialog without any over sampling of negative training data.

²⁰ <https://github.com/faneshion/MatchZoo>

Table 5: Comparison of different models over Ubuntu Dialog Corpus (UDC), MSDialog, and AliMe data sets. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.05$ measured by the Student's paired t-test.

Data Methods	UDC				MSDialog				AliMe			
	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2
BM25	0.6504	0.8206	0.5138	0.6439	0.4387	0.6329	0.2626	0.3933	0.6392	0.6407	0.2371	0.4204
BM25-PRF	0.6620	0.8292	0.5289	0.6554	0.4419	0.6423	0.2652	0.3970	0.6412	0.6510	0.2454	0.4209
ARC-II	0.6855	0.8978	0.5350	0.6959	0.5398	0.8662	0.3189	0.5413	0.7306	0.6595	0.2236	0.3671
MV-LSTM	0.6611	0.8936	0.4973	0.6733	0.5059	0.8516	0.2768	0.5000	0.7734	0.7017	0.2480	0.4105
DRMM	0.6749	0.8776	0.5287	0.6773	0.5704	0.9003	0.3507	0.5854	0.7165	0.6575	0.2212	0.3616
Duet	0.5692	0.8272	0.4756	0.5592	0.5158	0.8481	0.2934	0.5046	0.7651	0.6870	0.2433	0.4088
SMN	0.7327	0.9273	0.5948	0.7523	0.6188	0.8374	0.4529	0.6195	0.8145	0.7271	0.2881	0.4680
DMN	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673	0.7833	0.7629	0.3568	0.5012
DMN-KD	0.7655 ‡	0.9351 ‡	0.6443 ‡	0.7841 ‡	0.6728 ‡	0.9304 ‡	0.4908 ‡	0.7089 ‡	0.8323	0.7631	0.3596 ‡	0.5122 ‡
DMN-PRF	0.7719 ‡	0.9343 ‡	0.6552 ‡	0.7893 ‡	0.6792 ‡	0.9356 ‡	0.5021 ‡	0.7122 ‡	0.8435 ‡	0.7701 ‡	0.3601 ‡	0.5323 ‡

4.3 Evaluation Results

4.3.1 Performance Comparison on UDC and MSDialog.

We present evaluation results over different methods on UDC and MSDialog in Table 5. We summarize our observations as follows: (1) DMN-PRF model outperforms all the baseline methods including traditional retrieval models, deep text matching models and the state-of-the-art SMN model for response ranking on both conversation datasets. The results demonstrate that candidate response expansion with pseudo-relevance feedback could improve the ranking performance of responses in conversations. The main difference between DMN-PRF model and SMN model is the information extracted from retrieved feedback QA posts as external knowledge. This indicates the importance of modeling external knowledge with pseudo-relevant feedback beyond the dialog context for response selection. (2) DMN-KD model also outperforms all the baseline methods on MSDialog and UDC. These results show that the extracted QA correspondence matching knowledge could help the model select better responses. Comparing DMN-KD and DMN-PRF, their performances are very close. (3) If we compare the performances of DMN-PRF, DMN-KD with the degenerated model DMN, we can see that incorporating external knowledge via both pseudo-relevance feedback and QA correspondence knowledge distillation could improve the performance of the deep neural networks for response ranking with large margins. For example, the improvement of DMN-PRF against DMN on UDC is 4.83% for MAP, 1.60% for Recall@5, 8.19% for Recall@1, 5.11% for Recall@2 respectively. The differences are statistically significant with $p < 0.05$ measured by the Student's paired t-test.

4.3.2 Performance Comparison on AliMe Data. We further compare our models with the competing methods on AliMe data in Table 5. We find that: (1) our DMN model has comparable results in terms of MAP when compared with SMN, but has better Recall; (2) DMN-KD shows comparable or better results than all the baseline methods; (3) DMN-PRF significantly outperforms other competing baselines which shows the effectiveness of adding external pseudo-relevance feedback to the task; (4) both DMN-PRF and DMN-KD show better results than DMN, which demonstrates the importance of incorporating external knowledge via both pseudo-relevance feedback and QA correspondence knowledge distillation.

4.3.3 Performance Comparison over Different Response Types. We conduct fine-grained analysis on the performance of

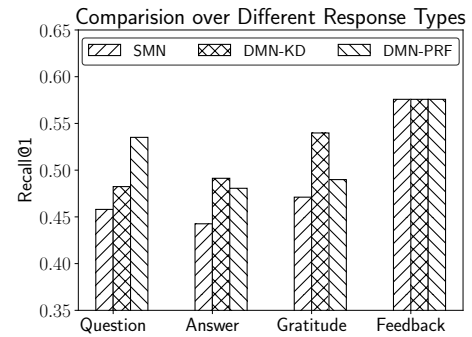


Figure 3: Performance comparison over different response types on MSDialog data.

different models on different response types. We annotated the user intents in 10,020 MSDialog utterances using Amazon Mechanical Turk²¹. We defined 12 user intent types including several types related to “questions” (original question, follow-up question, information request, clarifying question, and etc.), “answers” (potential answer and further details), “gratitude” (expressing thanks, greetings) and “feedback” (positive feedback and negative feedback). Then we trained a Random Forest classifier with TF-IDF features and applied this classifier to predict the response candidate types in the testing data of MSDialog. The dialog contexts were grouped by the type of the true response candidate. Finally we computed the average Recall@1 over different groups. Figure 3 shows the results. We find that both DMN-KD and DMN-PRF improve the performances of SMN for responses with type “questions”, “answers” and “gratitude”. This indicates that incorporating external knowledge with PRF or QA correspondence knowledge distillation can help the model select better responses, especially for QA related responses. For responses with type “Feedback”, DMN-KD and DMN-PRF achieved similar performances comparing with SMN.

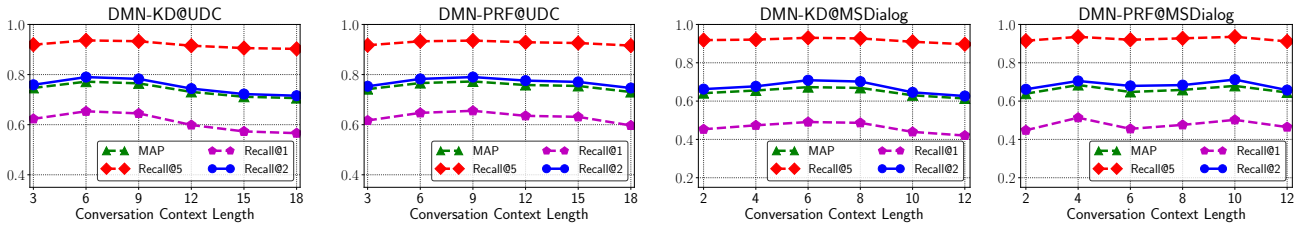
4.4 Model Ablation Analysis

We investigate the effectiveness of different components of DMN-PRF and DMN-KD by removing them one by one from the original

²¹<https://www.mturk.com/>

Table 6: Evaluation results of model ablation. “TB5” means the setting is the same with the results in Table 5. For DMN-KD, the model is the same with DMN if we remove M3. Numbers in bold font mean the result is better compared with other settings.

Model	Data	UDC				MSDialog			
		MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2
DMN-PRF	Only M1	0.7599	0.9294	0.6385	0.7761	0.5632	0.8509	0.3654	0.5579
	Only M2	0.7253	0.9271	0.5836	0.7440	0.4996	0.8584	0.2595	0.5021
	Inter-Dot (TB5)	0.7719	0.9343	0.6552	0.7893	0.6792	0.9356	0.5021	0.7122
	Inter-Cosine	0.7507	0.9260	0.6248	0.7675	0.6729	0.9356	0.4944	0.7027
	Inter-Bilinear	0.7228	0.9199	0.5829	0.7401	0.4923	0.8421	0.2647	0.4744
DMN-KD	Only M1	0.7449	0.9247	0.6167	0.7612	0.5776	0.8673	0.3805	0.5779
	Only M2	0.7052	0.9203	0.5538	0.7260	0.5100	0.8613	0.2794	0.5011
	Only M3	0.3887	0.6017	0.2015	0.3268	0.3699	0.6650	0.1585	0.2957
	M1+M2 (DMN)	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673
	M1+M3	0.7442	0.9251	0.6149	0.7612	0.6134	0.8860	0.4224	0.6266
	M2+M3	0.7077	0.9198	0.5586	0.7263	0.5141	0.8659	0.2885	0.5069
	Inter-Dot (TB5)	0.7655	0.9351	0.6443	0.7841	0.6728	0.9304	0.4908	0.7089
	Inter-Cosine	0.7156	0.9121	0.5770	0.7268	0.6916	0.9249	0.5241	0.7249
	Inter-Bilinear	0.7061	0.9135	0.5590	0.7225	0.4936	0.8224	0.2679	0.4814

**Figure 4: Performance of DMN-KD and DMN-PRF with different choices of conversation context length.**

model with UDC and MSDialog data. We also study the effectiveness of different interaction types for M1/M2/M3. Table 6 shows the results. We summarize our observations as follows: 1) For the interaction matrices, we find that the performance will drop if we remove any one of M1/M2 for DMN-PRF or M1/M2/M3 for DMN-KD. This indicates that all of word level interaction matching, sequence level interaction matching and external QA correspondence interaction matching are useful for response selection in information-seeking conversation. 2) For interaction types, we can find that dot product is the best setting on both UDC and MSDialog except the results of DMN-KD on MSDialog. The next best one is cosine similarity. Bilinear product is the worst, especially on MSDialog data. This is because bilinear product will introduce a transformation matrix A as an additional model parameter, leading to higher model complexity. Thus the model is more likely to overfit the training data, especially for the relatively small MSDialog data. 3) If we only leave one channel in the interaction matrices, we can find that M1 is more powerful than M2 for DMN-PRF. For DMN-KD, M1 is also the best one, followed by M2. M3 is the last one, but it still adds additional matching signals when it is combined with M1 and M2. The matching signals M3 from external collection could be supplementary features to the word embedding based matching matrix M1 and BiGRU representation based matching matrix M2.

4.5 Impact of Conversation Context Length

We further analyze the impact of the conversation context length on the performances of our proposed DMN-KD and DMN-PRF models. As presented in Figure 4, we find the performance first increases and then decreases, with the increase of conversation

Table 7: Examples of Top-1 ranked responses by different methods. y_i^k means the label of a response candidate.

Context	[User] I open Excel and it automatically formats my dates into American formatting. I have changed and saved the formatting to NZ style. However everytime I pull the document out of office 365 it reverts back to the American format. How do I stop this ? [Agent] Is it one file or all files in Excel ? [User] It does seem to be all Excel files. How do I change the global date format setting ?	
Method	y_i^k	Top-1 Ranked Response
SMN	0	Go to Settings ->System ->Tablet Mode....Change setting as indicated in the snapshot below.
DMN-KD	1	That is a Windows setting. Go to Control Panel >Regional settings. This will change date settings for all applications.
DMN-PRF	1	That is a Windows setting. Go to Control Panel >Regional settings. This will change date settings for all applications.

context length. The reason for these trends is that the context length controls the available previous utterances in the dialog context modeled by DMN-KD and DMN-PRF. If the context length is too small, there would be not enough information for the model to learn the matching patterns between the context and response candidates. However, setting the context length too large will also bring noise into the model results, since the words in utterances a few turns ago could be very different due to the topic changes during conversations.

4.6 Case Study

We perform a case study in Table 7 on the top ranked responses by different methods including SMN, DMN-KD and DMN-PRF. In this example, both DMN-KD and DMN-PRF produced correct top

ranked responses. We checked the retrieved QA posts by the correct response candidate and found that “*settings, regional, change, windows, separator, format, excel, panel, application*” are the most frequent terms. Among them “*excel*” is especially useful for promoting the rank of the correct response candidate, since this term which is included multiple times by the dialog context does not actually appear in the raw text of the correct response candidate. This gives an example of the effectiveness of incorporating external knowledge from the retrieved QA posts into response candidates.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a learning framework based on deep matching networks to leverage external knowledge for response ranking in information-seeking conversation systems. We incorporate external knowledge into deep neural models with pseudo-relevance feedback and QA correspondence knowledge distillation. Extensive experiments on both open benchmarks and commercial data show our methods outperform various baselines including the state-of-the-art methods. We also perform analysis on different response types and model variations to provide insights on model applications. For future work, we plan to model user intent in information-seeking conversations and learn meaningful patterns from user intent dynamics to help response selection. Incorporating both structured and unstructured knowledge into deep matching networks for response ranking is also interesting to explore.

6 ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1419693. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

- [1] J. Arguello, B. Choi, and R. Capra. 2017. Factors Affecting Users’ Information Requests. In *CAIR* '17.
- [2] A. Bordes, Y. Boureau, and J. Weston. 2017. Learning end-to-end goal-oriented dialog. *ICLR* '17.
- [3] G. Cao, J. Nie, J. Gao, and R. Stephen. 2008. Selecting Good Expansion Terms for Pseudo-relevance Feedback. In *SIGIR* '08.
- [4] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* (2014).
- [5] B. Dhingra, L. Li, X. Li, J. Gao, Y. Chen, F. Ahmed, and L. Deng. 2017. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. In *ACL* '17.
- [6] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM* '16.
- [7] B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NIPS* '14.
- [8] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM* '13.
- [9] Z. Ji, Z. Lu, and H. Li. 2014. An Information Retrieval Approach to Short Text Conversation. *CoRR* abs/1408.6988 (2014).
- [10] T. Kenter and M. de Rijke. 2017. Attentive memory networks: Efficient machine reading for conversational search. In *CAIR* '17.
- [11] D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* (2014).
- [12] V. Lavrenko and W. B. Croft. 2001. Relevance Based Language Models. In *SIGIR* '01.
- [13] F. Li, M. Qiu, H. Chen, X. Wang, X. Gao, J. Huang, J. Ren, Z. Zhao, W. Zhao, L. Wang, and G. Jin. 2017. AliMe Assist: An Intelligent Assistant for Creating an Innovative E-commerce Experience. In *CIKM* '17.
- [14] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and W. B. Dolan. 2016. A Persona-Based Neural Conversation Model. In *ACL* '16.
- [15] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *EMNLP* '16.
- [16] R. Lowe, N. Pow, I. Serban, and J. Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *CoRR* abs/1506.08909 (2015).
- [17] Y. Lv and C. Zhai. 2009. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *CIKM* '09.
- [18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS* '13.
- [19] B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW* '17.
- [20] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. 2016. Text Matching as Image Recognition. In *AAAI* '16.
- [21] M. Qiu, F. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *ACL* '17.
- [22] F. Radlinski and N. Craswell. 2017. A theoretical framework for conversational search. In *CHIIR* '17.
- [23] A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *ACL* '11.
- [24] S. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR* '94.
- [25] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.).
- [26] L. Shang, Z. Lu, and H. Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL* '15.
- [27] A. Sordani, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *NAACL* '15.
- [28] D. Spina, J. R. Trippas, L. Cavedon, and M. Sanderson. 2017. Extracting audio summaries to support effective spoken document search. *JAIST* '17 68, 9 (2017).
- [29] P. Thomas, D. McDuff, M. Czerwinski, and N. Craswell. 2017. MISC: A data set of information-seeking conversations. In *CAIR* '17.
- [30] Z. Tian, R. Yan, L. Mou, Y. Song, Y. Feng, and D. Zhao. 2017. How to Make Context More Useful? An Empirical Study on Context-Aware Neural Conversational Models. In *ACL* '17.
- [31] J. Trippas, D. Spina, M. Sanderson, and L. Cavedon. 2015. Towards understanding the impact of length in web search result summaries over a speech-only communication channel. In *SIGIR* '15.
- [32] O. Vinyals and Q. V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015).
- [33] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. In *AAAI* '16.
- [34] H. Wang, Z. Lu, H. Li, and E. Chen. 2013. A Dataset for Research on Short-Text Conversations. In *EMNLP* '13.
- [35] T. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P. Su, S. Ultes, and S. Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. *EACL* '17 (2017).
- [36] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL* '17.
- [37] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR* '17.
- [38] Z. Xu, B. Liu, B. Wang, C. Sun, and X. Wang. 2016. Incorporating Loose-Structured Knowledge into LSTM with Recall Gate for Conversation Modeling. *CoRR* (2016).
- [39] R. Yan, Y. Song, and H. Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*.
- [40] R. Yan, Y. Song, X. Zhou, and H. Wu. 2016. “Shall I Be Your Chat Companion?”: Towards an Online Human-Computer Conversation System. In *CIKM* '16.
- [41] R. Yan, D. Zhao, and W. E. 2017. Joint Learning of Response Ranking and Next Utterance Suggestion in Human-Computer Conversation System. In *SIGIR* '17.
- [42] L. Yang, Q. Ai, J. Guo, and W. B. Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM* '16.
- [43] L. Yang, H. Zamani, Y. Zhang, J. Guo, and W. B. Croft. 2017. Neural Matching Models for Question Retrieval and Next Question Prediction in Conversation. *CoRR* (2017).
- [44] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The Hidden Information State Model: A Practical Framework for POMDP-based Spoken Dialogue Management. *Comput. Speech Lang.* (2010).
- [45] J. Yu, M. Qiu, J. Jiang, J. Huang, S. Song, W. Chu, and H. Chen. 2018. Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce. *WSDM* '18.
- [46] H. Zamani, J. Dadashkarimi, A. Shakeri, and W. B. Croft. 2016. Pseudo-Relevance Feedback Based on Matrix Factorization. In *CIKM* '16.
- [47] C. Zhai and J. Lafferty. 2001. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM* '01.
- [48] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, and R. Yan. 2016. Multi-view Response Selection for Human-Computer Conversation. In *EMNLP*.