Learning to Expand: Reinforced Response Expansion for Information-seeking Conversations

Haojie Pan[1], Cen Chen[2†], Chengyu Wang[1], Minghui Qiu[1], Liu Yang[3], Feng Ji[1], Jun Huang[1]
[1] Alibaba Group [2] East China Normal University, China [3] University of Massachusetts at Amherst, China
{haojie.phj,chengyu.wcy,minghui.qmh,zhongxiu.jf,huangjun.hj}@alibaba-inc.com
cenchen@dase.ecnu.edu.cn,yangliuyx@gmail.com

ABSTRACT

Information-seeking conversation systems are increasingly popular in real-world applications, especially for e-commerce companies. To retrieve appropriate responses for users, it is necessary to compute the matching degrees between candidate responses and users' queries with historical dialogue utterances. As the contexts are usually much longer than responses, it is thus necessary to expand the responses (usually short) with richer information. Recent studies on pseudo-relevance feedback (PRF) have demonstrated its effectiveness in query expansion for search engines, hence we consider expanding response using PRF information. However, existing PRF approaches are either based on heuristic rules or require heavy manual labeling, which are not suitable for solving our task. To alleviate this problem, we treat the PRF selection for response expansion as a learning task and propose a reinforced learning method that can be trained in an end-to-end manner without any human annotations. More specifically, we propose a reinforced selector to extract useful PRF terms to enhance response candidates and a BERT-based response ranker to rank the PRF-enhanced responses. The performance of the ranker serves as a reward to guide the selector to extract useful PRF terms, which boosts the overall task performance. Extensive experiments on both standard benchmarks and commercial datasets prove the superiority of our reinforced PRF term selector compared with other potential soft or hard selection methods. Both case studies and quantitative analysis show that our model is capable of selecting meaningful PRF terms to expand response candidates and also achieving the best results compared with all baselines on a variety of evaluation metrics. We have also deployed our method on online production in an e-commerce company, which shows a significant improvement over the existing online ranking system.

CCS CONCEPTS

• **Applied computing** → *Electronic commerce*; • **Information systems** → *Retrieval models and ranking*.

† Cen Chen is the corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481932

KEYWORDS

Response expansion, Reinforcement learning, BERT, Info-seeking conversations

ACM Reference Format:
H. Pan, C. Chen, C. Wang, M. Qiu, L. Yang, F. Ji, J. Huang. 2021. Learning to Expand: Reinforced Response Expansion for Information-seeking Conversations. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3481932

1 INTRODUCTION

Intelligent personal assistant systems or chatbots such as Microsoft's XiaoIce, Apple's Siri, Google's Assistant have boomed during the last few years. It brings the interests from both academia and industry on information-seeking conversational systems, where end-users can access information with conversational interactions with the systems. Most of the assistant systems are based on retrieval-based methods as they can produce more informative, relevant, fluent, and controllable responses [15]. The common practice for these methods is to use historical dialogue utterances and the current user query as the "context" and use it to match the most relevant "response" from the candidates in databases [49, 50].

Recently, researchers have shown that external Pseudo-relevance Feedback (PRF) expansion is beneficial for context-response matching in multi-turn context understanding [50]. The goal of PRF expansion is to extract PRF terms from relevant documents to improve the original "query" representations. However, we find that it is also an important task to expand the "response" instead of the "query" for info-seeking conversations. The reasons are two-fold. First, the context is usually much longer than the response, which leads to a vocabulary gap between the context and the response. For example, as shown in Figure 1, some words in the context contain key information, such as the term "macros", which is relevant to "protected workbook" in the response but is not explicitly shown. Hence, it is important to expand the responses with richer information to better match the contexts. Second, to expand response is more efficient than query expansion for online retrieval systems as it requires no additional process of the input query, which makes it more suitable for real-world applications.

Despite the importance of response expansion for info-seeking conversations, few studies address this problem. To bridge this gap, this study seeks to examine the helpfulness of response expansion with external PRF information. The task is a non-trivial task that mainly has two challenges. First, existing studies [50] for PRF terms expansion are not directly applicable for the task, as these studies do not select high-quality PRF terms and simply use all the PRF

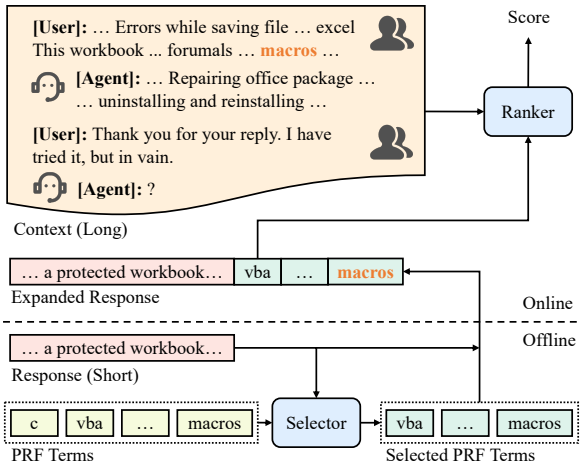

Figure 1: Overview of our proposed model "PRF-RL".

results as additional inputs. As shown in [2], some retrieved PRF terms are *unrelated* to the query, thus not helpful for the retrieval system. Secondly, after high-quality PRF terms are produced, it is challenging to model the interactions between the context, the response, and the PRF terms. Furthermore, it is a labor-intensive task to manually examine which PRF terms are suitable for a given response. There is a compelling need to automatically extract helpful PRF terms for response expansion without labeled signals.

To mitigate these challenges, we consider the response expansion problem as a learning task and build a effective way to guide the selection without explicitly labeled signals. The overview of our framework is outlined in Figure 1, which contains a PRF term selector for response expansion and a ranker for response ranking. Firstly, we consider Reinforcement Learning (RL) for our task, due to its effectiveness in exploration with implicit feedback [4, 5, 40]. We model the problem as a Markov Decision Process (MDP) and propose a general framework to jointly learn a *reinforced selector* with a *response ranker*. We treat the reinforced PRF term selector as the agent that takes the actions to select a subset of PRF terms based on the state representation. These selected PRF terms will be further fed into the ranking module together with the conversation context and the response. The ranking module and the validation data play a role of an environment and output rewards for those selection actions. The resulting rewards can guide the reinforced selector to generate higher-quality PRF terms to improve the ranking results. Secondly, we adopt BERT [3] with different input formats as the encoder of our response ranker, since the language models pre-trained on large unsupervised corpora [16, 25] has demonstrated stunning success in NLP. After training, the selector can generate the response expansion offline and save them in databases, hence the efficiency of the online ranker will not be affected. We refer to our proposed whole model as "**PRF-RL**".

We conduct extensive experiments on both public and industrial datasets, i.e., the MSDialog dataset [24] that contains customer service dialogs from Microsoft Answers community and the AliMe-CQA dataset [15] collected from the chat logs between the customers and the service assistant agent in an e-commerce APP. We compare our method with various baselines, such as traditional retrieval models, neural ranking models, a strong multi-turn conversation response ranking baseline [57] and pre-trained language model based methods. Our method outperforms all the baseline methods on a variety of evaluation metrics. Besides, to demonstrate the effectiveness of our reinforced PRF term selector, we also compare our model with other potential PRF selection methods, including the rule-based selection method mentioned in [50], soft and hard selection by gate functions such as `tanh` or Gumbel softmax trick [11]. Both the numerical results and case studies show the superiority of our proposed reinforced selector. We have deployed our proposed "**PRF-RL**" model in an online information-seeking system on a real e-commerce production AliMe. The online A/B test results show that our model significantly outperforms the existing online ranking system.

In a nutshell, our contributions can be summarized as follows:

(1) Our study is one of the earliest attempts to analyze how response expansion can be better utilized to improve BERT-based retrieval models. Response expansion is more efficient for online retrieval systems than query expansion, as it requires no additional process of the input query, which makes it more suitable for building real-world applications.

(2) We tackle the response expansion problem from a learning perspective and propose a novel method, i.e., PRF-RL, for the problem. Our method consists of two modules: i) a reinforced selector to extract useful PRF terms, and ii) a BERT response ranker with PRF. To the best of our knowledge, it is the first work that employs an RL-based strategy to select high-quality PRF terms for response ranking in the information-seeking systems.

(3) Experimental results on both public and industrial datasets show that our methods outperform various baselines and show that our reinforced PRF terms selector is superior to other competing PRF term selection mechanisms. We have also deployed our method on an online chatbot system in an e-commerce company. [1]

2 RELATED WORK

We summarize the related works on conversational information-seeking systems, query expansion, reinforcement learning and neural response ranking.

2.1 Conversational Information-seeking Systems and Query Expansion

Our research is relevant to conversational information-seeking systems. Radlinski and Craswell [26] described the basic features of conversational information-seeking systems. Thomas et al. [35] released the Microsoft Information-Seeking Conversation (MISC) dataset, which contains information-seeking conversations with a human intermediary. Zhang et al. [55] introduced the System Ask User Respond (SAUR) paradigm for conversational search and recommendation. In addition to conversational search models, researchers have also studied the medium of conversational search. Spina et al. [33] studied the ways of presenting search results over speech-only channels to support conversational search [33, 36].

In conversational information-seeking systems, the context can be longer than the candidate responses. Hence, it is necessary to expand the responses. In IR systems, pseudo-relevance feedback

[1] The source code of our method will be released in EasyTransfer [23].

(PRF) has been demonstrated the effectiveness for query expansion [2, 13, 17, 29, 53, 54]. Cao et al. [2] incorporates multiple manual features to identify useful expanding terms. Lv and Zhai [17] compares methods for estimating query language models with pseudo-relevance feedback in ad-hoc information retrieval. Li et al. [14] proposed an end-to-end neural model to make the query directly interacts with the retrieved documents. The idea of query expansion using PRF inspires us to use the candidate response as a query to retrieve relevant terms, thus boosting ranking performance by reducing the problem of vocabulary mismatch between the context and the original response candidates.

2.2 Reinforcement Learning

Reinforcement Learning (RL) is a series of goal-oriented algorithms that have been studied for many decades in many disciplines [1]. The recent development in deep learning has greatly contributed to this area and has delivered amazing achievements in many domains, such as playing games against humans [32]. There are two lines of work in RL: value-based methods and policy-based methods. Value-based methods, including SARSA [30] and the Deep Q Network [19], take actions based on estimations of expected long-term return. On the other hand, policy-based methods such as REINFORCE [42] optimize for a strategy that can map states to actions that promise the highest reward. It is proved that reinforcement learning is effective in data selection problems over many areas, such as active learning [5], co-training [43], and other applications of supervised learning [6, 40]. Our proposed reinforced PRF term selector is trained by REINFORCE.

2.3 Neural Response Ranking

There is growing interest in research about conversation response generation and ranking with deep learning and reinforcement learning [7]. There are two main categories of the previous works, including retrieval-based methods [34, 45, 47, 48, 50, 56] and generation-based methods [22, 27, 31, 38]. Our research work is related to retrieval-based methods. There has been some research on response ranking in multi-turn conversations with retrieval-based methods. Yang et al. [50] studied how to integrate external knowledge into deep neural networks for response ranking in information-seeking conversations. [57] investigated matching a response with conversation contexts with dependency information learned by attention mechanisms of Transformers. The model proposed in this paper incorporating a reinforced PRF terms selection mechanism to select meaningful PRF terms to boost the performance of pre-trained model based response ranking model. Recently, language models pre-trained on massive unsupervised corpora [3, 12, 16, 21, 25, 52] has achieved a significant improvement in many natural language processing tasks, ranging from syntactic parsing to natural language inference [3, 21], as well as machine reading comprehension [3, 46], information retrieval tasks [20, 51]. The pre-trained models such as BERT [3] are also applied to response selection [9, 37, 41]. Our model applies BERT as a part of the response ranker with the response expanded by selected PRF terms.

3 OUR APPROACH

In this section, we formally present our PRF-RL model. We begin with a brief problem definition, followed by the proposed method.

3.1 Problem Definition

We formulate the problem of response ranking with pseudo-relevance feedback as follows. Given an information-seeking conversation dataset $\mathcal{D} = \{(\mathcal{U}_i, r_i, y_i)\}_{i=1}^{N}$, where $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ is a m-turn dialog context, $u_i = \{w_{i,1}^{(u)}, w_{i,2}^{(u)}, ..., w_{i,L_{u_i}}^{(u)}\}$ as the utterance in the i-th turn of this dialog and L_{u_i} is the number of sub-tokens of this utterance. $r = \{w_1^{(r)}, w_2^{(r)}, ..., w_n^{(r)}\}$ is a response candidate and $y \in \{0, 1\}$ is the corresponding label. When the psudo-relevance feedback information is incorporated, for each response r, we have PRF term set $\mathcal{P} = \{p_1, p_2, ..., p_k\}$ and $p_i = \{w_{i,1}^{(p)}, w_{i,2}^{(p)}, ..., w_{i,L_{p_i}}^{(p)}\}$ is i-th PRF term and and L_{p_i} is the number of sub-tokens of this PRF term. The task is to learn a model which has two sub-modules: (1) a selection module $f(\cdot)$ to select a meaningful subset $\mathcal{P}' \subseteq \mathcal{P}$ of PRF terms and (2) a ranking module $g(\cdot)$ with $\mathcal{D}$ and $\mathcal{P}'$ as inputs to rank the responses. Given $\mathcal{U}$ and $\mathcal{P}$, the model should be able to generate a prediction $\hat{y}$ for response r for ranking with other candidate responses.

3.2 Model Overview

In the following sections, we describe our proposed model for response ranking with pseudo-relevance feedback. Given a set of retrieved PRF term candidates [2], we first propose a reinforced PRF term selector to select a subset of the PRF terms, and feed them into a BERT-based response ranking model, which outputs the final predictions. We will first briefly introduce how the BERT response ranker works and then leave more space to introduce our proposed reinforced PRF term selector in detail. Figure 2 presents an overview of the proposed method.

3.3 BERT Response Ranker

As mentioned above, once we have a context $\mathcal{U}$, a response r and a set of PRF terms $\mathcal{P}$, we can concatenate them with the following specific format as the input of BERT $\mathbf{x} = \{$ `[CLS]`, u_1, `[EOT]`, u_2, `[EOT]`, ..., u_m, `[SEP]`, r, `[SEP]`, p_1, `[SEP]` p_2, `[SEP]`, ..., $p_k\}$. Here we use `[EOT]` to separate the multiple turns of the dialog and use `[SEP]` to separate different PRF terms since they are independent with each other. After multiple standard BERT layers, we can get a contextual representation $T_{\texttt{[CLS]}}$ of `[CLS]` token. We then feed the contextual representation into an extra feed forward network with sigmoid activation function to predict the ranking score of r given $\mathcal{U}$ and $\mathcal{P}$, illustrated as follows:

$$T_{\texttt{[CLS]}} = \texttt{BERT}(\mathbf{x}). \tag{1}$$

$$\hat{y} = g(\mathcal{U}, \mathcal{P}, r) = \sigma(\mathbf{W}_o^T T_{\texttt{[CLS]}} + \mathbf{b}). \tag{2}$$

The response ranking model can be optimized by gradient descent based methods and the cross-entropy loss is applied as follows.

[2]The details of how the candidates are retrieved and examples are shown in the experiment section.

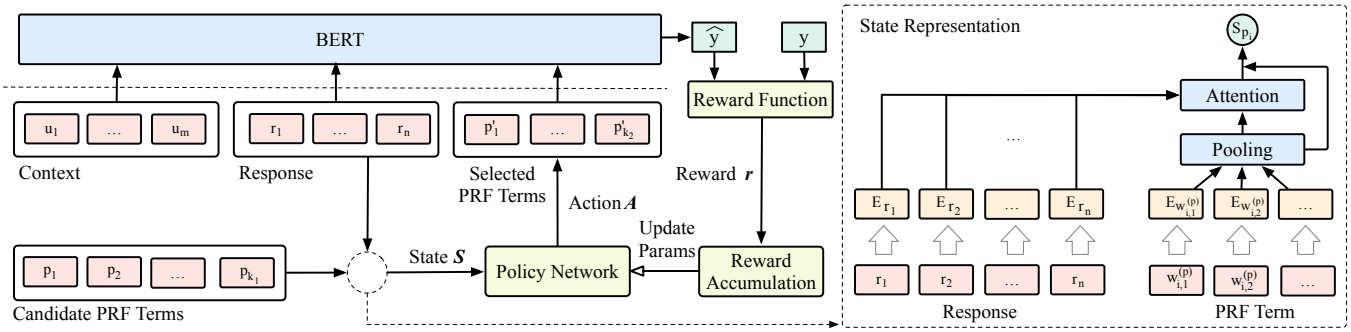

Figure 2: Reinforced pseudo-relevance feedback selector.

$$Loss = -\left(\sum_{i=1}^{N} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)\right) \tag{3}$$

3.4 Reinforced PRF Term Selector

The PRF term selection process can be viewed as a sequential decision-making problem and modeled as a Markov Decision Process, which can be further solved by reinforcement learning. Under the reinforcement setting, the PRF term selector serves as the *agent* and interacts with the environment consisting of the BERT response ranker and a utility evaluation dataset. With the help of a learnable policy network, the agent takes the *actions* of selecting or dropping a given PRF term, where the decision is based on a *state representation* of each candidate term. Then the BERT response ranker takes the selected PRF terms as one of the inputs and further provides *rewards* for guiding the agent. The overview of the reinforced PRF term selector is shown in Figure 2.

To be specific, we formulate the learning framework as follows. Given a response r and the corresponding PRF terms set $\mathcal{P} = \{p_i\}_{i=1}^{k_1}$, where k_1 is the number of PRF terms before selection. We can apply a state representation network $z(r, p_i)$ for each PRF term p_i to obtain states $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{k_1}$ and feed each $\mathbf{s}_i$ into the policy network $\pi(\mathbf{s}_i)$. According to the policy, the agent takes the corresponding actions $\mathcal{A} = \{a_1, a_2, ..., a_{k_1}\}$, where $a_i \in \{0, 1\}$. The selected subset of PRF terms $\mathcal{P}' = \{p_i | \forall p_i \in \mathcal{P}, a_i = 1\}$, together with $\mathcal{U}$ and r, are then fed into the response ranker to output a prediction. The parameters of the response ranker will be updated with label y and a reward $\mathbf{r}$ will be produced after evaluating the ranker's performance on a held-out validation dataset.

3.4.1 State representation. The state $\mathbf{s}_i = z(r, p_i)$ of given response r and a PRF term p_i is a d-dimension continuous real valued vector. We use the attention mechanism between token embeddings of r and p_i to obtain a contextual vector as follows.

With the WordPiece embeddings [44], we can represent tokens of r and p_i as:

$$\mathbf{E_r} = \{\mathbf{e}_{w_1^{(r)}}, \mathbf{e}_{w_2^{(r)}}, ..., \mathbf{e}_{w_n^{(r)}}\}, \tag{4}$$

$$\mathbf{E_{p_i}} = \{\mathbf{e}_{w_{i,1}^{(p)}}, \mathbf{e}_{w_{i,2}^{(p)}}, ..., \mathbf{e}_{w_{i,L_{p_k}}^{(p)}}\}. \tag{5}$$

We first use pooling methods, e.g., max-pooling, to get a d-dimension vector $\mathbf{h_i} = pooling(\mathbf{E_{p_i}})$ to represent the PRF term and obtain the attended weighted context as follows:

$$\mathbf{h}'_i = \sum_{j=1}^{n} \frac{exp(\mathbf{h}_i \cdot \mathbf{e}_{w_j^{(r)}})}{\sum_{k=1}^{n} exp(\mathbf{h}_i \cdot \mathbf{e}_{w_k^{(r)}})} \mathbf{e}_{w_j^{(r)}}. \tag{6}$$

Finally we get the state representation as $\mathbf{s}_i = \mathbf{h}_i + \mathbf{h}'_i$.

3.4.2 Policy Network and Actions. The agent (i.e., the reinforced selector) takes actions to decide whether to select the PRF term p_i as a part of the inputs for the BERT response ranker. The action a_i is sampled [3] according to a probability distribution produced by the policy network $\pi(\mathbf{s_i})$ that consists of one feed-forward network as follows:

$$\pi(\mathbf{s_i}) = P(a_i|\mathbf{s_i}) = softmax(\mathbf{W}_\pi{}^T \mathbf{s}_i + \mathbf{b}_\pi), \tag{7}$$

where $\mathbf{W}_\pi$ and $\mathbf{b}_\pi$ are trainable parameters of the policy network.

3.4.3 Reward Function. After the selection, we have a subset of PRF terms $\mathcal{P}'$, which is used to update the ranker together with $\mathcal{U}$, r and y. With the selected PRF terms and the updated BERT response ranker, we can conduct inference on the validation data to compute the rewards. More specifically, for each batch, the reward function is defined as a delta loss on the validation data:

$$\mathbf{r}_b = \mathcal{L}_{b-1} - \mathcal{L}_b, \tag{8}$$

where $\mathcal{L}_b$ and $\mathcal{L}_{b-1}$ are the cross-entropy losses on the validation data for the current batch and the previous batch respectively. The intuition is that if the current loss is smaller than the previous loss, we encourage the selector to follow the current policy by giving it a positive reward. Other metrics generated on the validation data could also be incorporated. To enable fast training, we obtain the reward by evaluating a subset of the validation data. This subset is referred to as the *reward set*, which is randomly sampled from the validation data and changed at the end of every episode.

Furthermore, we compute the future total reward for each batch after an episode since the decisions of the agent not only have a direct impact on the immediate rewards but also have long-term influence. The amended reward $\mathbf{r}'$ is formulated as follows.

$$\mathbf{r}'_b = \sum_{k=0}^{N_e - b} \gamma^k \mathbf{r}_{b+k}, \tag{9}$$

[3] In testing phase, a_i is chosen by the maximum probability.

where N_e is the number of batches in this episode, $\mathbf{r}'_b$ is the future total reward for batch b, and γ is the reward discount factor.

3.4.4 Optimization. We use a policy gradient method, i.e., REINFORCE [42] to optimize our proposed reinforced PRF Term selector. For a given episode, our goal is to maximize the expected total reward, which can be formulated as follows:

$$J(\Theta) = E_{\pi_\Theta}[\sum_{b=1}^{N_e} \mathbf{r}_b], \tag{10}$$

where the policy network π_Θ is parameterized by Θ. The policy network can be updated by the gradient as follows:

$$\Theta \leftarrow \Theta + \alpha \frac{1}{B} \sum_{i=1}^{B} \mathbf{r}_i \nabla_\Theta log \pi_\Theta(\mathbf{S}_i). \tag{11}$$

Here, α is the learning rate, B is the batch size.

3.5 Training Process

The selector and the ranker modules are learned jointly as they interact with each other closely during training. For each batch, the PRF term selector selects a subset of PRF terms $\mathcal{P}'$ from the input PRF term set $\mathcal{P}$. Then the BERT response ranker uses $\mathcal{P}'$ together with the context $\mathcal{U}$ and the response r as inputs to output the predictions. To optimize the BERT response ranker, we use a standard gradient descent method to minimize the loss function in Eq. 3. The reinforced PRF selector intervenes before every iteration of the ranker update by selecting helpful PRF terms to augment the candidate responses. Such an intervention process has a direct impact on the gradient computed for the ranker update. The BERT response ranker provides a reward in turn to evaluate the utility of the selected PRF terms. After each episode, the policy network of the selector is updated with the policy gradient algorithm with the stored (state, action, reward) triples. A detailed description of our algorithm is shown in Algorithm 1

Algorithm 1 Training Procedure

Require: Training data $\mathcal{D}_{train} = \{\mathcal{X}_i\}_{i=1}^{N} = \{(\mathcal{U}_i, r_i, \mathcal{P}_i, y_i)\}_{i=1}^{N}$;
Validation data $\mathcal{D}_{val} = \{\mathcal{X}_i\}_{i=1}^{N'} = \{(\mathcal{U}_i, r_i, \mathcal{P}_i, y_i)\}_{i=1}^{N'}$;
Episode N_e, validation sample rate q;

```
Initialize the pre-trained BERT response ranker g(·);
Initialize the policy network π_Θ as the PRF term selector;
for episode l = 1 to N_e do
    Obtain the random batch sequence D'_train = {X_b}_{b=1}^B;
    Obtain the reward set D_reward by random sampling from
    D_val with rate q;
    for each X_b in {X_b}_{b=1}^B do
        Obtain state S_b by z(r_b, P_b);
        Sample action A_b according to policy π(S_b);
        Obtain PRF subset P'_b according to A_b;
        Update the ranker g(·) by X'_b = {U_b, r_b, P'_b, y_b};
        Obtain the reward r_b on D_reward;
        Store (S_b, A_b, r_b) to an episode history H;
    end for
    for each (S_b, A_b, r_b) in H do
        Obtain the future total reward r'_b as in Eq. 10;
        Update the policy network π_Θ following Eq. 11;
    end for
    Empty H;
end for
```

4 EXPERIMENTS

Table 1: The statistics of experimental datasets, where C denotes context and R denotes response. # Cand. per C denotes the number of candidate responses per context. Note that we didn't filter any stop words or words with low frequency when we computed the average length of contexts or responses.

Data	MSDialog			AliMe-CQA		
Items	Train	Valid	Test	Train	Valid	Test
# C-R pairs	173k	37k	35k	32k	3.9k	4k
# Cand. per C	10	10	10	16.1	15	15.1
# + Cand. per C	1	1	1	6.0	5.2	4.8
Min # turns per C	2	2	2	2	2	2
Max # turns per C	11	11	11	2	2	2
Avg # turns per C	5.0	4.9	4.4	2	2	2
Avg # words per C	451	435	375	17.9	18.2	18.4
Avg # words per R	106	107	105	6.0	5.2	4.8

In this section, we conduct extensive experiments to evaluate the performance of the proposed framework. We also present the online deployment results to show its superiority.

4.1 Dataset Description

We evaluate our method and the competing methods on two info-seeking conversation datasets: MSDialog dataset and AliMe-CQA dataset as used in [50]. The data split and the statistics of the data is shown in Table 1.

4.1.1 MSDialog. The MSDialog[4] dataset is a labeled dialog dataset of question answering (QA) interactions between information seekers and answer providers from an online forum on Microsoft products [24]. Previous works [50] have a pre-processed version that is suitable for experimenting with conversation response ranking models. The ground truth responses returned by the real agents are the positive response candidates, and negative sampling has been adopted to create nine negative response candidates for each context query. We only removed some common prefixes such as "<<AGENT>>:" and use WordPiece [44] to tokenize the context and the response candidates for further modeling.

4.1.2 AliMe-CQA Data. We collected the multi-turn question answering chat logs between customers and a chatbot from the AliMe conversation system [5]. This chatbot is built based on a question-to-question matching system [15], where for each query, it finds the most similar candidate question in a QA database and returns its answer as the reply. To form an information-seeking conversation QA dataset, we firstly select more than 3k multi-turns context to

[4]https://ciir.cs.umass.edu/downloads/msdialog/
[5]https://www.alixiaomi.com/

form queries and apply this conversation system to retrieve the top-15 most similar candidate questions as the "response" in our setting. A group of business analysts is asked to annotate the candidate "response". If the "response" is similar to the input query (context), the label will be positive, otherwise negative. In the process of annotation, if the confidence score of answering a given query (context) is low, the system will prompt three top related questions(response candidates) for users to choose from. We collected such user click logs as our external data, where we treat the clicked question as positive and the others as negative. We have recalled about 50k context-response pairs from this annotation process and remove all of the contexts that have zero positive candidate responses. The language of the context and response is Chinese and we use character-level tokenization for further modeling.

4.2 PRF Term Candidates Retrieving

The goal of pseudo-relevance feedback (PRF) is to extract terms from the top-ranked documents during the retrieval process to help discriminate relevant documents from the irrelevant ones [2].

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 maximum IDF weights) in feedback documents. Given the retrieved top K_1 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 K_2 terms from θ as the PRF terms for the response candidate. We first use the response candidate as the query to retrieve top K_1 QA posts from the external corpus with BM25 as the source of the external knowledge for each response candidate. While retrieving relevant documents, we perform several preprocessing steps including tokenization, punctuation removal, and stop word removal for the query. We set $K_1 = 10$ for MSDialog dataset, and $K_1 = 15$ for AliMe dataset. The external retrieval source corpus for MSDialog and AliMe are Stack Overflow data and unlabeled AliMe QA data respectively.

Once the response-relevant QA posts are retrieved for each response, we count the term frequencies for all terms in these posts and select the top K_2 frequent terms as the PRF terms for each response candidate. K_2 is set as 10 for both the MSDialog dataset and the AliMe dataset.

4.3 Experimental Setup

4.3.1 Baselines. We explore different baselines lying on four categories, including traditional retrieval models, neural ranking models, a strong multi-turn conversation response ranking method, pre-trained language model based models as follows:

BM25 [28] is a traditional retrieval model, which uses the dialog context as the query to retrieve response candidates for response ranking.

ARC-II [10], **MV-LSTM** [39], **DRMM** [8], **DUET** [18] are neural ranking models proposed in recent years for ad-hoc retrieval and question answering. MV-LSTM is a representation focused model and ARC-II, DRMM are interaction focused models. Duet is a hybrid method of both representation focused and interaction-focused models.

DAM [57] is a strong baseline model for response ranking in multi-turn conversations. DAM also represents and matches a response with its multi-turn context using dependency information learned by Transformers.

BERT-Ranker is a general classification framework proposed in BERT [3] paper. It uses `[SEP]` and segment embedding to separate the query and answer and incorporates a pre-trained language model for contextual representation. The predictions are based on the contextual vector of `[CLS]` token.

PRF-RL is our proposed model consisting of a reinforced PRF term selector and a BERT response ranker.

4.3.2 Evaluation Methodology. For evaluation metrics of both MSDialog and AliMe-CQA, we adopted mean average precision (MAP) and Recall@k which is the recall at top k ranked responses from n available candidates for a given conversation context. Following previous related works [57], here we reported Recall@1, Recall@2, and Recall@5 on both two datasets. For AliMe-CQA, we reported an extra metric Precision@1 for further exploration, since there are multiple positive candidates of a given query. One should also notice that for the MSDialog dataset, the value of precision@1 is equal to the recall@1 since there is only one positive candidate for each query in this dataset.

4.3.3 Experimental Settings. The four neural ranking models are experimented using the MatchZoo [6] toolkit. We use the code [7] released by [57] to tune the DAM model on our datasets. We use the Hugging Face transformer version [8] of the BERT model to implement BASE-BERT classifier. We choose the $\textbf{BERT}_{\textbf{BASE}}$ (L=12, H=768, A=12) as our pre-trained BERT encoder for both BERT-Ranker and PRF-RL. The models are implemented in PyTorch and run on 1 Tsela P100 GPU. We present the detailed hyper-parameters used for experiments as follows.

For the MSDialog dataset, the context length is truncated by 384 and the response length is truncated by 96. The batch size is set to 12. We use Adam optimizer with linear decay for both two models. The learning rate for BERT-Ranker is set to 3e-5 following the previous works. For PRF-RL, we firstly pre-trained the BERT response ranker without PRF term selection of learning rate 3e-5 for 1000 steps, and then jointly trained the reinforced PRF term selector and the BERT response ranker with learning rate 1e-4, 1.5e-5 respectively. For reinforcement learning, we use max-pooling in the state representation, the number of the episode is set to 100, the reward discount reward factor is set to 0.3, and the reward set is randomly sampled from 0.5% of the validation dataset considering the trade-off of quality and efficiency of training.

For the AliMe dataset, the context length is truncated by 100 and the response length is truncated by 50. The batch size is set to 32. We again use Adam optimizer with linear decay for both two models. The learning rate for BERT-Ranker is set to 3e-5. For PRF-RL, we firstly pre-trained the BERT response ranker with a learning rate 3e-5 for 200 steps, and then jointly trained the reinforced PRF term selector and the BERT response ranker with a learning rate 1e-4, 5e-6 respectively. We again use max-pooling in the state representation, the number of the episode is set to 10, the reward discount reward

[6] https://github.com/NTMC-Community/MatchZoo
[7] https://github.com/baidu/Dialogue/tree/master/DAM
[8] https://github.com/huggingface/transformers

Table 2: Comparison of different models over MSDialog and eCommerce data sets. Numbers in bold font mean the result is the best compared with other models.

Data	MSDialog				AliMe-CQA				
Methods	Recall@1	Recall@2	Recall@5	MAP	Precision@1	Recall@1	Recall@2	Recall@5	MAP
BM25 [28]	0.2626	0.3933	0.6329	0.4387	0.5811	0.2012	0.3201	0.5378	0.6310
ARC-II [10]	0.3189	0.5413	0.8662	0.5398	0.6075	0.1717	0.3027	0.6190	0.6841
MV-LSTM [39]	0.2768	0.5000	0.8516	0.5059	0.5925	0.1657	0.3194	0.6015	0.6813
DRMM [8]	0.3507	0.5854	0.9003	0.5704	0.6868	0.2194	0.3563	0.6036	0.7048
Duet [18]	0.2934	0.5046	0.8481	0.5158	0.6679	0.1920	0.3408	0.6302	0.7162
DAM [57]	0.7012	0.8527	0.9715	0.8150	0.7558	0.2472	0.3969	0.6919	0.7773
BERT-Ranker	0.7667	0.8926	**0.9852**	0.8580	0.8476	0.2968	0.4622	0.7263	0.8513
PRF-RL	**0.7872**	**0.9032**	0.9792	**0.8700**	**0.8717**	**0.3181**	**0.4868**	**0.7576**	**0.8675**

factor is set to 0.3, and the reward set is randomly sampled from 1% of the validation dataset.

4.4 Comparison with Baselines

We present evaluation results over different methods on MSDialog and AliMe-CQA in Table 2.

4.4.1 Performance Comparison on MSDialog. From the results on the MSDialog dataset, we have the following findings. First, the transformer-based models (DAM, BERT-Ranker, PRF-RL) show significant improvements compared with traditional retrieval models and other neural ranking models, which further proves the powerful representation capabilities of the Transformer. Second, compared with the DAM model, when the pre-trained language model BERT is incorporated, the performance also has a good improvement. Last but not least, our model performs the best over all the other baselines on Recall@1, Recall@2, and MAP, which we can see that incorporating external knowledge via pseudo-relevance feedback could improve the performance of the BERT-based response ranking models by large margins. Specifically, compared with BERT-Ranker, our proposed PRF-RL model has a comparable result in terms of Recall@5, but an improvement of 2.05% for Recall@1, 1.06% for Recall@2, 1.20% for MAP. This shows the benefits of considering PRF selection.

4.4.2 Performance Comparison on AliMe-CQA. After comparing our PRF-RL model with other baselines on the AliMe-CQA dataset in Table 2. We find those similar findings as using MSDialog. First, our model achieves the best performance against all the baselines in terms of all evaluation metrics. Specifically, for precision-based metrics, our model achieves 2.41% and 1.62% improvements compared with the strongest baseline BERT-Ranker in terms of Precision@1 and MAP; And for recall-based metrics, our model achieves improvements of 2.13% for Recall@1, 2.46% for Recall@2 and 3.13% for Recall@5. Second, Compared with the MSDialog dataset, the absolute values of Recall@k are lower. This phenomenon comes from multiple positive candidates given one query, in such case, a lower recall comparing with MSDialog dataset does not necessarily mean the method has lower performance. In practice, for info-seeking conversation systems, Precision@1 is the most important metric as only the top-1 response will be returned to the customer. In this metric, all the methods on AliMe-CQA tend to have better performance than MSDialog.

In all, our proposed method has a clear advantage over all the competing methods in both datasets, which demonstrates the usefulness of our method for info-seeking conversations.

4.5 Comparison with Other PRF Methods

To further explore how well our reinforced PRF term selector contributes to the overall model performance, we build several baseline methods that use different ways to incorporate the information of pseudo-relevance feedbacks. Here we have the following baselines:

RULE-PRF is a simple method to feed the PRF terms filtered by term frequencies, which is introduced in [50].

PRF-ML-Tanh is a soft selection method which outputs a score $q_{p_i} = \tanh(\mathbf{s}_i)$. Before the embedding of PRF terms p_i is feed into BERT, the embedding will first be scaled by this score q_{p_i}. It is more like a gate function to ensure that the gradients can be back-propagated into the selector.

PRF-ML-Sig is a similar soft selection method which replaces the *tanh* function with the *sigmoid* function.

PRF-ML-Gumb is a method that can not only produce a hard selection decision but also can be optimized by gradient descent based algorithms. It uses a categorical re-parameterization trick with the Gumbel softmax function [11] that enables the model to sample discrete random variables in a way that is differentiable.

The experimental results are shown in Table 3. By exploring the results, we have the following findings:

• Overall, the incorporation of pseudo-relevance feedback can improve the performance of the BERT-based response ranking models. The RULE-PRF method without selection achieves improvements of 0.46% for Recall@1 on the MSDialog dataset and 0.34% for Recall@1 on the AliMe-CQA dataset. However, in terms of some metrics such as Recall@2 and Recall@5 on MSDialog dataset, and Precision@1 on the AliMe-CQA dataset, adding PRF terms can hurt the ranking model's performance. This means the necessity to implement better ways for PRF term selection, instead of the simple approach.

• For a soft version of machine learning based PRF term selection models, the tanh gating is better than sigmoid gating. Hard version (PRF-ML-Gumb) of machine learning based PRF term selection performance better in terms of Recall@2, Recall@5 on MSDialog Dataset and Recall@1 on AliMe-CQA dataset, which can be

Table 3: Comparison of different models over MSDialog and AliMe-CQA datasets. Numbers in bold font mean the result is the best compared with other models. Here "P" means precision and "R" means recall. The Precision@1 results on the MSDialog dataset are omitted since it is equal to Recall@1.

MSDialog					
Methods	P@1	R@1	R@2	R@5	MAP
BERT-Ranker	/	0.7667	0.8926	**0.9852**	0.8580
RULE-PRF	/	0.7713	0.8906	0.9826	0.8601
PRF-ML-Tanh	/	0.7770	0.8886	0.9815	0.8626
PRF-ML-Sigmoid	/	0.7736	0.8906	0.9823	0.8607
PRF-ML-Gumbel	/	0.7719	0.8946	0.9823	0.8614
PRF-RL	/	**0.7872**	**0.9032**	0.9792	**0.8700**
AliMe-CQA					
Methods	P@1	R@1	R@2	R@5	MAP
BERT-Ranker	0.8476	0.2968	0.4622	0.7263	0.8513
RULE-PRF	0.8460	0.3002	0.4785	0.7422	0.8523
PRF-ML-Tanh	0.8604	0.3060	0.5027	0.7631	0.8654
PRF-ML-Sigmoid	0.8340	0.3041	0.4835	0.7362	0.8549
PRF-ML-Gumbel	0.8566	0.3084	0.4781	0.7429	0.8554
PRF-RL	**0.8717**	**0.3181**	**0.4868**	**0.7576**	**0.8675**

concluded that the hard selection has potential to achieve better performance but training hard selection by machine learning is not straightforward and intuitive.

• After incorporating the hard selection by our reinforced PRF term selector, our proposed PRF-RL model achieves the best performance against outperforms all the PRF selection methods. This observation again proves the effectiveness of our framework in PRF term selection.

4.6 Hyper-parameter Sensitivity Analysis

In this section, we first examine the performance of PRF-RL with different choices of the number of episode steps on the MSDialog dataset. As in Figure 3 (Left), we have these observations. First, we find that our method is generally insensitive to this parameter, although by setting the number of episode steps as 100, our method has a slight improvement. Second, we observe that the performance in terms of Recall@5 is inconsistent with other metrics such as Recall@1 and Recall@2. A good performance in Recall@5 may not come with good performances in Recall@1 and Recall@2. This means the model optimizes the metrics differently, and may not achieve the best performance on all these metrics. In practice, Recall@1 is more important than Recall@2 or Recall@5. Thus we can set the episode step as 100 as we can achieve the best Recall@1, Recall@2, and MAP. We then proceed to examine our model performance w.r.t. the future reward discount factors on the MSDialog dataset in Figure 3 (Right). In general, the model performance is not very sensitive to this parameter as well, although our method has slightly better performance in terms of Recall@1 and Recall@2 by setting the reward discount factor as 0.3. From Figure 3, we find our method is pretty robust as it is not very sensitive to these

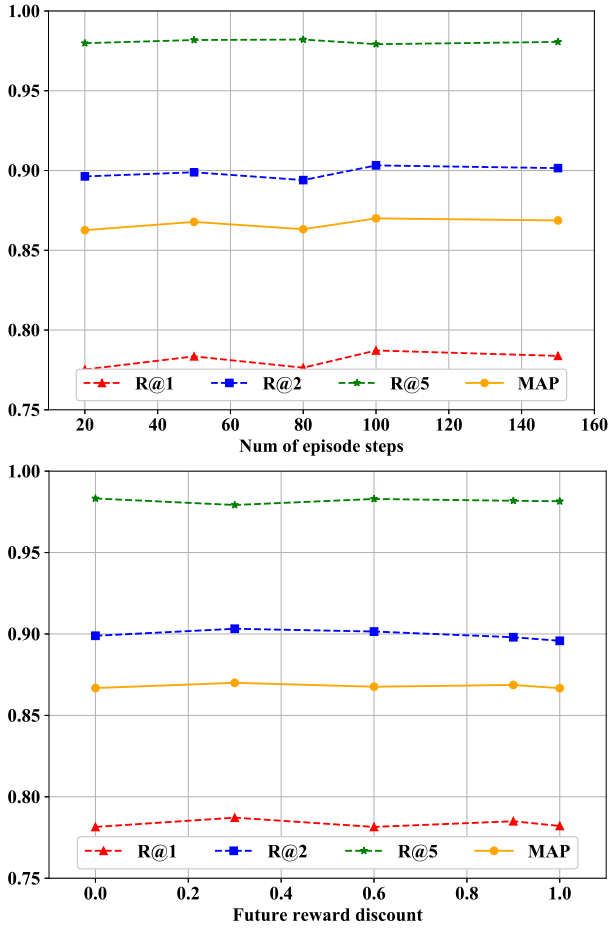

Figure 3: Performance of PRF-RL with different choices of numbers of episodes and discount factors over the test partition of MSDialog. The reward discount factor is set to 0.3 when we compare different choices of the number of episodes (Left). The number of episodes is set to 100 while we compare different choices of the discount factors (Right).

parameter settings. But still, by conducting combining the findings in both Figures, we can find a relatively better combination of hyper-parameters for our method.

4.7 Case Study

One of the most interesting features of our model is that it exhibits a certain level of interpretability as the selection process is explicit. We then present a case study to examine whether the generated PRF terms for expanding the response candidates are meaningful.

Figure 4 shows one of the cases generated from the MSDialog dataset. In this case, the user wants to edit and save a "excel notebook" which seems to be protected by multiple formulas and macros. The agent proposes a general solution to the user but not working. So the response might be other solutions or ask the user about more detailed descriptions of the problem he has met. Since the "excel notebook" can be converted to "protected" one through

Context	[USER]: Errors where detected while saving [**file** name]. Microsoft **Excel** may be able to save the **file** by removing or repairing some features. To make the repairs in a new **file**, click Continue. To cancel saving the **file**, click Cancel. When I try to continue it did not save at all. this **Excel workbook** has multiple forumals and **macros** and its protected, I dont own it. But I need to work on it and save. I have tried everything and **using** Office 365 in windows 10. Please help. [AGENT]: Good Day My Name is Phillip Roos and i will be happy to help the best way possible and in a timely manner. Please try the following solution of repairing your office package: The first thing you can do is to navigate and select the start button 2) Type in control panel 3) Navigate to ""Programs"". 4) Then select ""Uninstall a Program"" 5) Then find your office installation and right click (Usually say Microsoft Office 2016 or something similar) 6) Select change 7) Now a window will pop asking for ""Quick Repair"" or ""Online Repair"" 8) Select the Quick Repair option and reboot 9) If the problem continues select the Online Repair option and reboot If the problem continues i do recommend uninstalling and reinstalling the office 2016/2013 package. Please let me know if this helped or not, i will happily assist you further. Kind Regards Phillip Roos. [USER]: Thank for your reply. I have tried it , but in vain.
Response	Good Day as i gathered from your question you want to open and edit a protected **excel workbook** and save changes to it ?
ML-Tanh	excel vba workbook c active file using saveas macros userform
ML-Sigmoid	excel vba workbook c active file using saveas macros userform
ML-Gumbel	excel vba workbook c active file using saveas macros userform
RL	excel vba workbook c active file using saveas macros userform

Figure 4: Examples of PRF selection using different models. Here, the response is a positive candidate and we use different levels of transparency of green color to demonstrate the levels of PRF term selection for expanding this response. The bold orange terms are the PRF terms shown in context/response, sorted by their term frequencies.

some "macros", the contextual correlation between "protected excel notebook" with "macros" and "vba" is strong. From the results, we can find the soft selections of ML-Tanh and ML-Sigmoid can confuse the ranking model since they both give more weights on irrelevant terms such as "c" and "active". ML-Gumbel has the same problem, it selects a general term "using" and dropped the "excel" and "workbook" which are relevant to the response. Our PRF-RL model achieve the best result since it can (1) select the exact match terms such as "excel" and "workbook" in the response, (2) avoid selecting irrelevant or general terms such as "c", "active" and "using", (3) select contextually correlated terms such as "macros", which appears in the context but not in the response, which means it can be used to improve the recall of the response candidate. In all, encouragingly we find the selection process made by the proposed PRF-RL model is insightful and intuitive.

4.8 Online A/B Test

Finally, we deployed the proposed PRF-RL model on an online chatbot engine called AliMe [9] in the e-commerce company Alibaba, and conduct an A/B test on our proposed model and the existing online ranking system without considering external PRF expansion. In the AliMe chatbot engine, for each user query or request, it employs a two-stage model to find a suitable candidate response. It first calls back at most 15 candidate responses from tens of thousands of candidates and then uses a neural ranker to rerank the candidate responses. The engine uses both of the two systems, one is our method and the other is the online method, to rerank the candidates. Note that the online method is a degenerated version of our method without considering PRF expansion. Overall we have randomly selected 13,549 conversational QA-pairs. After filtering out all the context queries that have zero positive responses in the call-back set, we have collected 325 context queries for each system. We then ask a customer agent to annotate the results of both methods. We obtain the number of the hit of the top-1 ranked candidates and compared the precision@1 score.

As a result, our proposed PRF-RL model has Precision@1 of 67.69%, which has a significant relative improvement (12.24%) compared with the existing online ranking system (60.3%). Such improvement is considered to be a big improvement for the chatbot engine. This further shows the advantage of our proposed method and the usefulness of the external PRF expansion. For efficiency, the candidate responses are expanded in a pre-processing step which is done in an offline manner, thus it ensures zero efficiency drop.

5 CONCLUSION

In this work, we propose a principled approach to automatically select useful pseudo-relevance feedback terms to help information-seeking conversations. The proposed method considers a reinforced selector to interact with a BERT response ranker to generate high-quality pseudo-relevance feedback terms, and the performance of the ranker can help to guide the behaviors of the selector. Extensive experiments on both public and industrial datasets show our model outperforms the competing models. We have also deployed our proposed model in AliMe chatbot and observe a large improvement over the existing online ranking system.

[9] https://www.alixiaomi.com/

REFERENCES

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* 34, 6 (2017), 26–38.

[2] G. Cao, J. Nie, J. Gao, and R. Stephen. 2008. Selecting Good Expansion Terms for Pseudo-relevance Feedback. In *SIGIR '08*.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. [n.d.]. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[4] Yang Fan, Fei Tian, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2017. Learning What Data to Learn. *CoRR* (2017).

[5] Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to Active Learn: A Deep Reinforcement Learning Approach. *CoRR* (2017).

[6] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement Learning for Relation Classification From Noisy Data. In *AAAI*. AAAI Press, 5779–5786.

[7] J. Gao, M. Galley, and L. Li. 2018. Neural Approaches to Conversational AI. *CoRR* abs/1809.08267 (2018).

[8] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM '16*.

[9] Matthew Henderson, Ivan Vulic, Daniela Gerz, Iñigo Casanueva, Pawel Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksic, and Pei-Hao Su. 2019. Training Neural Response Selection for Task-Oriented Dialogue Systems. In *ACL*. 5392–5404.

[10] B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NIPS '14*.

[11] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *CoRR* (2019).

[13] V. Lavrenko and W. B. Croft. 2001. Relevance Based Language Models. In *SIGIR '01*.

[14] Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. In *EMNLP*.

[15] 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*.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692

[17] Y. Lv and C. Zhai. 2009. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *CIKM '09*.

[18] B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW '17*.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[20] Rodrigo Nogueira and Kyunghyun Cho. [n.d.]. Passage Re-ranking with BERT. *CoRR* ([n. d.]).

[21] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[22] 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*.

[23] Minghui Qiu, Peng Li, Chengyu Wang, Haojie Pan, An Wang, Cen Chen, Xianyan Jia, Yaliang Li, Jun Huang, Deng Cai, and Wei Lin. 2021. EasyTransfer - A Simple and Scalable Deep Transfer Learning Platform for NLP Applications. *CIKM 2021* (2021). https://arxiv.org/abs/2011.09463

[24] C. Qu, L. Yang, W. B. Croft, J. R. Trippas, Y. Zhang, and M. Qiu. 2018. Analyzing and Characterizing User Intent in Information-seeking Conversations. In *SIGIR '18*. 989–992.

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[26] F. Radlinski and N. Craswell. 2017. A theoretical framework for conversational search. In *CHIIR '17*.

[27] A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *ACL '11*.

[28] S. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR '94*.

[29] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.).

[30] G. A. Rummery and M. Niranjan. 1994. *On-Line Q-Learning Using Connectionist Systems*. Technical Report TR 166. Cambridge University Engineering Department, Cambridge, England.

[31] L. Shang, Z. Lu, and H. Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL '15*.

[32] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550 (Oct. 2017), 354–.

[33] 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).

[34] C. Tao, W. Wu, C. Xu, W. Hu, D. Zhao, and R. Yan. 2019. Multi-Representation Fusion Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *WSDM '19*.

[35] P. Thomas, D. McDu, M. Czerwinski, and N. Craswell. 2017. MISC: A data set of information-seeking conversations. In *CAIR '17*.

[36] 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*.

[37] Jesse Vig and Kalai Ramea. 2019. Comparison of Transfer-Learning Approaches for Response Selection in Multi-Turn Conversations. (2019).

[38] O. Vinyals and Q. V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015).

[39] 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*.

[40] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. [n.d.]. R^3: Reinforced Ranker-Reader for Open-Domain Question Answering. In *AAAI*. 5981–5988.

[41] Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2019. Domain Adaptive Training BERT for Response Selection. *CoRR* (2019).

[42] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* 8 (1992), 229–256.

[43] Jiawei Wu, Lei Li, and William Yang Wang. [n.d.]. Reinforced Co-Training. In *NAACL-HLT*. 1252–1262.

[44] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, and Jeffrey Dean et al. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* (2016).

[45] 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*.

[46] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. [n.d.]. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *NAACL-HLT*. 2324–2335.

[47] R. Yan, Y. Song, and H. Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*.

[48] 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*.

[49] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware Response Ranking with Transformers in Information-seeking Conversation Systems. In *WWW*. 2592–2598.

[50] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *SIGIR '18*.

[51] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *CoRR* (2019).

[52] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*. 5754–5764.

[53] H. Zamani, J. Dadashkarimi, A. Shakery, and W. B. Croft. 2016. Pseudo-Relevance Feedback Based on Matrix Factorization. In *CIKM '16*.

[54] C. Zhai and J. Lafferty. 2001. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM '01*.

[55] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *CIKM '18*.

[56] 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*.

[57] X. Zhou, L. Li, D. Dong, Y. Liu, Y. Chen, W. X. Zhao, D. Yu, and H. Wu. 2018. Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. In *ACL '18*.