NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval

Canjia Li[1], **Yingfei Sun**[1], **Ben He**[1,3✉], **Le Wang**[1,4], **Kai Hui**[2],
Andrew Yates[5], **Le Sun**[3], **Jungang Xu**[1]

[1] University of Chinese Academy of Sciences, Beijing, China [2] SAP SE, Berlin, Germany
[3] Institute of Software, Chinese Academy of Sciences, Beijing, China
[4] Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
[5] Max Planck Institute for Informatics, Saarbrücken, Germany

{licanjia17, wangle315}@mails.ucas.ac.cn, {yfsun, benhe✉, xujg}@ucas.ac.cn
kai.hui@sap.com, ayates@mpi-inf.mpg.de, sunle@iscas.ac.cn

Abstract

Pseudo relevance feedback (PRF) is commonly used to boost the performance of traditional information retrieval (IR) models by using top-ranked documents to identify and weight new query terms, thereby reducing the effect of query-document vocabulary mismatches. While neural retrieval models have recently demonstrated strong results for ad-hoc retrieval, combining them with PRF is not straightforward due to incompatibilities between existing PRF approaches and neural architectures. To bridge this gap, we propose an end-to-end neural PRF framework that can be used with existing neural IR models by embedding different neural models as building blocks. Extensive experiments on two standard test collections confirm the effectiveness of the proposed NPRF framework in improving the performance of two state-of-the-art neural IR models.

1 Introduction

Recent progress in neural information retrieval models (NIRMs) has highlighted promising performance on the ad-hoc search task. State-of-the-art NIRMs, such as DRMM (Guo et al., 2016), HiNT (Fan et al., 2018), (Conv)-KNRM (Xiong et al., 2017; Dai et al., 2018), and (Co)-PACRR (Hui et al., 2017, 2018), have successfully implemented insights from traditional IR models using neural building blocks. Meanwhile, existing IR research has already demonstrated the effectiveness of incorporating relevance signals from top-ranked documents through pseudo relevance feedback (PRF) models (Buckley and Robertson, 2008; Diaz et al., 2016). PRF models expand the query with terms selected from top-ranked documents, thereby boosting ranking performance by reducing the problem of vocabulary mismatch between the original query and documents (Rocchio, 1971). Existing neural IR models do not have a mechanism for treating expansion terms differently from the original query terms, however, making it non-trivial to combine them with existing PRF approaches. In addition, neural IR models differ in their architectures, making the development of a widely-applicable PRF approach a challenging task.

To bridge this gap, we propose a generic neural pseudo relevance feedback framework, coined NPRF, that enables the use of PRF with existing neural IR models. Given a query and a target document, the top-ranked documents from the initial ranking are consumed by NPRF, which expands the query by interpreting it from different perspectives. Given a target document to evaluate, NPRF produces a final relevance score by considering the target document's relevance to these top-ranked documents and to the original query.

The proposed NPRF framework can directly incorporate different established neural IR models, which serve as the concrete scorers in evaluating the relevance of a document relative to the top-ranked documents and to the query, without changing their architectures. We instantiate the NPRF framework using two state-of-the-art neural IR models, and we evaluate their performance on two widely-used TREC benchmark datasets for ad-hoc retrieval. Our results confirm that the NPRF framework can substantially improve the performance of both models. Moreover, both neural models perform similarly inside the NPRF framework despite the fact that without NPRF one model performed substantially worse than the other model. The contributions of this work are threefold: 1) the novel NPRF framework; 2) two instantiations of the NPRF framework using two state-of-the-art neural IR models; and 3) the experiments that confirm the effectiveness of the NPRF framework.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4482–4491
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

The rest of this paper is organized as follows. Section 2 presents the proposed NPRF framework in details. Following that, Section 3 describes the setup of the evaluation, and reports the results. Finally, Section 4 recaps existing literature, before drawing conclusions in Section 5.

2 Method

In this section, we introduce the proposed neural framework for pseudo relevance feedback (NPRF). Recall that existing unsupervised PRF models (Rocchio, 1971; Lavrenko and Croft, 2001; Ye et al., 2009) issue a query to obtain an initial ranking, identify promising terms from the top-m documents returned, and expand the original query with these terms. Rather than selecting the expanded terms within the top-m documents, NPRF uses these documents directly as expansion queries by considering the interactions between them and a target document. Thus, each document's ultimate relevance score depends on both its interactions with the original query and its interactions with these feedback documents.

2.1 Overview

Given a query q, NPRF estimates the relevance of a target document d relative to q as described in the following steps. The architecture is summarized in Figure 1. Akin to the established neural IR models like DRMM (Guo et al., 2016), the description is based on a query-document pair, and a ranking can be produced by sorting the documents according to their scores.

- **Create initial ranking.** Given a document corpus, a ranking method $rel_q(q, d)$ is applied to individual documents to obtain the top-m documents, denoted as D_q for q.

- **Extract document interactions.** To evaluate the relevance of d, each d_q in D_q is used to expand q, where d is compared against each d_q, using a ranking method $rel_d(d_q, d)$.

- **Combine document interactions.** The relevance scores $rel_d(d_q, d)$ for individual $d_q \in D_q$ are further weighted by $rel_q(q, d_q)$, which serves as an estimator for the confidence of the contribution of d_q relative to q. The weighted combination of these relevance scores is used to produce a relevance score for d, denoted as $rel_D(q, D_q, d)$.

While the same ranking model can be used for both $rel_q(., .)$ and $rel_d(., .)$, we denote them separately in the architecture. In our experiments, the widely-used unsupervised ranking method BM25 (Robertson et al., 1995) serves as $rel_q(., .)$; meanwhile two state-of-the-art neural IR relevance matching models, namely, DRMM (Guo et al., 2016) and K-NRM (Xiong et al., 2017), serve as the ranking method $rel_d(., .)$. However, it is worth noting that in principle rel_q and rel_d can be replaced with any ranking method, and the above choices mainly aim to demonstrate the effectiveness of the NPRF framework.

2.2 Model Architecture

The NPRF framework begins with an initial ranking for the input query q determined by $rel_q(., .)$, which forms D_q, the set of the top-m documents D_q. The ultimate query-document relevance score $rel_D(q, D_q, d)$ is computed as follows.

Extracting document interactions. Given the target document d and each feedback document $d_q \in D_q$, $rel_d(., .)$ is used to evaluate the relevance between d and d_q, resulting in m real-valued relevance scores, where each score corresponds to the estimated relevance of d according to one feedback document d_q.

As mentioned, two NIRMs are separately used to compute $rel_d(d_q, d)$ in our experiments. Both models take as input the cosine similarities between each pair of terms in d_q and d, which are computed using pre-trained word embeddings as explained in Section 3.1. Given that both models consider only unigram matches and do not consider term dependencies, we first summarize d_q by retaining only the top-k terms according to their *tf-idf* scores, which speeds up training by reducing the document size and removing noisy terms. In our pilot experiments, the use of top-k *tf-idf* document summarization did not influence performance. For different $d_q \in D_q$, the same model is used as $rel_d(., .)$ for different pairs of (d_q, d) by sharing model weights.

Combining document interactions. When determining the relevance of a target document d, there exist two sources of relevance signals to consider: the target document's relevance relative to the feedback documents D_q and its relevance relative to the query q itself. In this step, we combine $rel_d(d_q, d)$ for each $d_q \in D_q$ into an overall feedback document relevance score

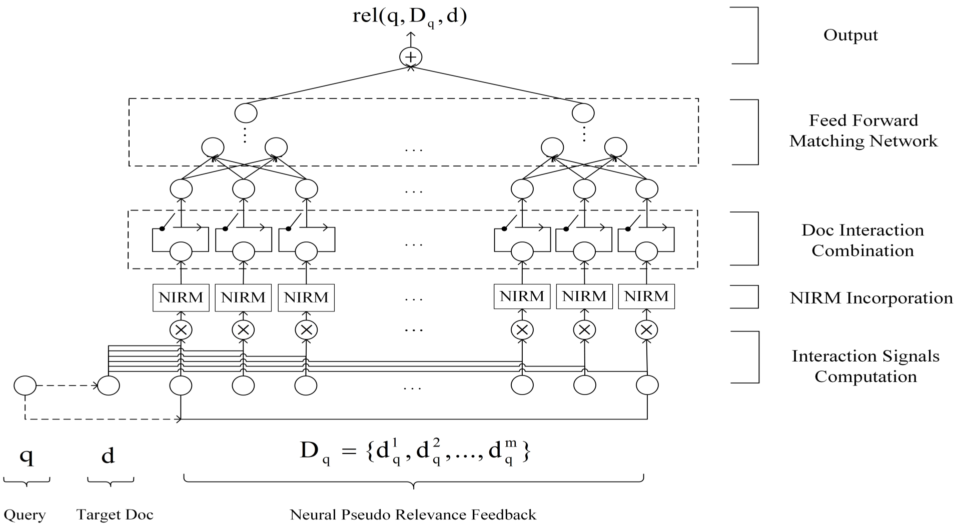
rel(q, D_q, d)
Output
Feed Forward Matching Network
Doc Interaction Combination
NIRM
NIRM
NIRM
NIRM
NIRM
NIRM
NIRM Incorporation
Interaction Signals Computation
q
d
D_q = {d_q^1, d_q^2, ..., d_q^m}
Query
Target Doc
Neural Pseudo Relevance Feedback

Figure 1: Architecture of the proposed neural pseudo relevance feedback (NPRF) framework.

$rel_D(q, D_q, d)$. When combining the relevance scores, the agreement between q and each d_q is also important, since d_q may differ from q in terms of information needs. The relevance of d_q from the initial ranking $rel_q(q, d_q)$ is employed to quantify this agreement and weight each $rel_d(d_q, d)$ accordingly.

When computing such agreements, it is necessary to remove the influence of the absolute ranges of the scores from the initial ranker. For example, ranking scores from a language model (Ponte and Croft, 1998) and from BM25 (Robertson et al., 1995) can differ substantially in their absolute ranges. To mitigate this, we use a smoothed min-max normalization to rescale $rel_q(q, d_q)$ into the range $[0.5, 1]$. The min-max normalization is applied by considering $min(rel_q(q, d_q)|d_q \in D_q)$ and $max(rel_q(q, d_q)|d_q \in D_q)$. Hereafter, $rel_q(q, d_q)$ is used to denote this relevance score after min-max normalization for brevity. The (normalized) relevance score is smoothed and then weighted by the relevance evaluation of d_q, producing a weighted document relevance score $rel_d{}'(d_q, d)$ for each $d_q \in D_q$ that reflects the relevance of d_q relative to q. This computation is described in the following equation.

$$rel_d{}'(d_q, d) = rel_d(d_q, d)(0.5 + 0.5 \times rel_q(q, d_q)) \quad (1)$$

As the last step, we propose two variants for combining the $rel_d{}'(d_q, d)$ for different d_q into a single score $rel_D(q, D_q, d)$: (i) performing a direct summation and (ii) using a feed forward network with a hyperbolic tangent ($tanh$) non-linear activation. Namely, the first variant simply sums up the scores, whereas the second takes the ranking positions of individual feedback documents into account.

2.3 Optimization and Training

Each training sample consists of a query q, a set of m feedback documents D_q, a relevant target document d^+ and a non-relevant target document d^- according to the ground truth. The Adam optimizer (Kingma and Ba, 2014) is used with a learning rate 0.001 and a batch size of 20. Training normally converges within 30 epochs, with weights uniformly initialized. A hinge loss is employed for training as shown below.

$$loss(q, D_q, d^+, d^-) =$$
$$max(0, 1 - rel(q, D_q, d^+) + rel(q, D_q, d^-))$$

3 Evaluation

3.1 Evaluation Setup

Dataset. We evaluate our proposed NPRF framework on two standard test collections, namely, TREC1-3 (Harman, 1993) and Robust04 (Voorhees, 2004). TREC1-3 consists of 741,856 documents with 150 queries used in the TREC 1-3 ad-hoc search tasks (Harman, 1993, 1994, 1995). Robust04 contains 528,155 documents and 249 queries used in the TREC 2004 Robust track (Voorhees, 2004). We use those two collections to balance between the number of queries and the TREC pooling depth, i.e., 100 on both collections, allowing for sufficient training data. Manual relevance judgments are available on both collections, where both the relevant and non-relevant documents are labeled for each query.

Two versions of queries are included in our experiments: a short keyword query (*title query*), and a longer description query that restates the corresponding keyword query's information need in terms of natural language (*description query*). We evaluate each type of query separately using the metrics Mean Average Precision at 1,000 (MAP), Precision at 20 (P@20) (Manning et al., 2008), and NDCG@20 (Järvelin and Kekäläinen, 2002).

Preprocessing. Stopword removal and Porter's stemmer are applied (Manning et al., 2008). The word embeddings are pre-trained based on a pool of the top 2,000 documents returned by BM25 for individual queries as suggested by (Diaz et al., 2016). The implementation of Word2Vec[1] from (Mikolov et al., 2013) is employed. In particular, we employ CBOW with the dimension set to 300, window size to 10, minimum count to 5, and a subsampling threshold of 10^{-3}. The CBOW model is trained for 10 iterations on the target corpus.

Unsupervised ranking models serve as baselines for comparisons. We use the open source Terrier platform's (Macdonald et al., 2012) implementation of these ranking models:

- **BM25** (Robertson et al., 1995), a classical probabilistic model, is employed as an unsupervised baseline. The hyper-parameters b and k_1 are tuned by grid search. As mentioned in Sec. 2.1, BM25 also generates the initial rankings D_q, serving as $rel_q(.,.)$ in the NPRF framework.

- On top of BM25, we use an adapted version of Rocchio's query expansion (Ye et al., 2009), denoted as **BM25+QE**. Note that, as demonstrated in the results, BM25+QE's performance is comparable with the base neural IR models, including DRMM, K-NRM and PACRR. This illustrates the difficulty in making improvements on the TREC benchmarks through the uses of deep learning methods. The hyper-parameters, including the number of feedback documents and the number of expansion terms, are optimized using grid search on training queries.

- In addition, **QL+RM3**, the query likelihood language model with the popular RM3 PRF (Lavrenko and Croft, 2001), is used as another unsupervised baseline.

Neural IR models are used for $rel_d(.,.)$. As mentioned in Section 2.1, two unigram neural IR models are employed in our experiments:

- **DRMM**. We employ the variant with the best effectiveness on Robust04 according to (Guo et al., 2016), namely, DRMM$_{LCH\times IDF}$ with the original configuration.

- **K-NRM**. Due to the lack of training data compared with the commercial data used by (Xiong et al., 2017), we employ a K-NRM variant with a frozen word embedding layer. To compensate for this substantial reduction in the number of learnable weights, we add an additional fully connected layer to the model. These changes lead to a small but competitive K-NRM variant, as demonstrated in (Hui et al., 2018).

- We additionally implement **PACRR** (Hui et al., 2017) for the purpose of performing comparisons, but do not use PACRR to compute $rel_d(.,.)$ due to the computational costs. In particular, PACRR-firstk is employed where the first $1,000$ terms are used to compute the similarity matrices, and the original configuration from (Hui et al., 2017) is used.

- **NIRM(QE)** uses the modified query generated by the query expansion of BM25+QE (Ye et al., 2009) as input to the neural IR model. Both DRMM and K-NRM are used to instantiate NIRM(QE).

- **Variants of the proposed NPRF approach.** As indicated in Section 2.2, NPRF includes two variants that differ in the combination of the relevance scores from different $d_q \in D_q$: the variant **NPRF$_{\mathbf{ff}}$** uses a feed forward network with a hidden layer with five neurons to compute $rel(d, D_q)$, and the other variant **NPRF$_{\mathbf{ds}}$** performs a direct summation of the different relevance scores. For the purposes of comparison, we additionally introduce another variant coined **NPRF$_{\mathbf{ff'}}$**, where the relevance of d_q to q is not considered in the combination by directly setting $rel'_d(d, d_q) = rel_d(d, d_q)$ in place of Equation 1, thereafter combining the scores with a fully connected layer as in NPRF$_{ff}$. We combine each of the three NPRF variants with

[1] https://code.google.com/p/word2vec/

the DRMM and K-NRM models, and report results for all six variants. Our implementation of the NPRF framework is available to enable future comparisons[2].

Akin to (Guo et al., 2016; Xiong et al., 2017; Hui et al., 2017), the NIRM baselines and the proposed NPRF are employed to re-rank the search results from BM25. In particular, the top-10 documents from the unsupervised baseline are used as the pseudo relevance feedback documents D_q as input for NPRF, where each $d_q \in D_q$ is represented by its top-20 terms with the highest tf-idf weights. As illustrated later in Section 3.3, NPRF's performance is stable over a wide range of settings for both parameters.

Cross-validation. Akin to (Hui et al., 2018), owing to the limited number of labeled data, five-fold cross-validation is used to report the results by randomly splitting all queries into five equal partitions. In each fold, three partitions are used for training, one for validation, and one for testing. The model with the best MAP on the validation set is selected. We report the average performance on all *test* partitions. A two-tailed paired t-test is used to report the statistical significance at 95% confidence interval.

3.2 Results

Comparison to BM25. We first compare the proposed NPRF models with the unsupervised BM25. The results are summarized in Tables 1 and 2, where the best result in each column is highlighted in bold. From Tables 1 and 2, it can be seen that the proposed NPRF variants obtain significant improvement relative to BM25 on both test collections with both kinds of test queries. Moreover, the results imply that the use of different query types does not affect the effectiveness of NPRF, which consistently outperforms BM25.

Comparison to neural IR models. NPRF is further compared with different neural IR models, as summarized in Tables 3 & 4. It can be seen that NPRF regularly improves on top of the NIRM baselines. For both types of queries, NPRF-DRMM outperforms DRMM and NPRF-KNRM outperforms K-NRM when re-ranking BM25. Remarkably, the proposed NPRF is able to improve the weaker NIRM baseline. For instance, on Robust04, when using the description queries, DRMM and K-NRM obtain highly different results, with MAPs of 0.2630 and 0.1687 after reranking the initial results from BM25, respectively. When NPRF is used in conjunction with the NIRM models, however, the gap between the two models is closed; that is, MAP=0.2801 for NRFF$_{ds}$-DRMM and MAP=0.2800 for NRFF$_{ds}$-KNRM (see Table 4). This finding highlights that our proposed NPRF is robust with respect to the use of the two embedded NIRM models. A possible explanation for the poor performance of K-NRM on two TREC collections is the lack of training data, as suggested in (Dai et al., 2018). While K-NRM could be improved by introducing weak supervision (Dai et al., 2018), we achieve the same goal by incorporating pseudo relevance feedback information without extra training data.

While the six NPRF variants exhibit similar results across both kinds of queries, NPRF$_{ds}$-DRMM in general achieves the best performance on Robust04, and NPRF$_{ds}$-KNRM appears to be the best variant on TREC1-3. In the meantime, NPRF$_{ds}$ outperforms NPRF$_{ff}$ variants. One difference between the two methods is that NPRF$_{ff}$ considers the position of each d_q in the D_q ranked documents, whereas NPRF$_{ds}$ simply sums up the scores regardless of the positions. The fact that NPRF$_{ds}$ performs better suggests that the ranking position within the D_q documents may not be a useful signal. In the remainder of this paper, we mainly report on the results obtained by NPRF$_{ds}$.

Comparison to query expansion baselines. In Table 5, the proposed NPRF model is compared with three kinds of query expansion baselines, namely, the unsupervised BM25+QE (Ye et al., 2009), QL+RM3 (Lavrenko and Croft, 2001), and DRMM/K-NRM(QE), the neural IR models using expanded queries as input. According to Table 5, the unsupervised BM25+QE baseline appears to achieve better performance in terms of MAP@1k, owing to its use of query expansion to match relevant documents containing the expansion terms from the whole collection. On the other hand, NPRF$_{ds}$, which reranks the top-1000 documents returned by BM25, outperforms the query expansion baselines in terms of early precision, as measured by either NDCG@20 or P@20. These measures on shallow rankings are particularly important for general IR applications where the quality of the top-ranked results is crucial to the user satisfaction. Moreover, our NPRF outperforms

[2] https://github.com/ucasir/NPRF

	Title						Description					
Model	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
BM25	0.2408	-	0.4803	-	0.4947	-	0.2094	-	0.4613	-	0.4838	-
NPRF$_{ff}$-DRMM	0.2669†	10.85%	0.5010	4.31%	0.5119	3.47%	0.2509†	19.80%	0.5257†	13.95%	0.5393†	11.46%
NPRF$_{ff'}$-DRMM	0.2671†	10.93%	0.5023†	4.59%	0.5116	3.42%	0.2504†	19.58%	0.5163†	11.93%	0.5291†	9.37%
NPRF$_{ds}$-DRMM	0.2698†	12.03%	0.5187†	7.99%	0.5282†	6.77%	**0.2527**†	20.67%	**0.5283**†	14.53%	0.5444†	12.52%
NPRF$_{ff}$-KNRM	0.2633†	9.34%	0.5033	4.80%	0.5171	4.52%	0.2486†	18.71%	0.5240†	13.59%	0.5398†	11.58%
NPRF$_{ff'}$-KNRM	0.2654†	10.22%	0.5077†	5.70%	0.5216†	5.44%	0.2462†	17.60%	0.5197†	12.65%	0.5363†	10.84%
NPRF$_{ds}$-KNRM	**0.2707**†	12.41%	**0.5303**†	10.42%	**0.5406**†	9.29%	0.2505†	19.61%	0.5270†	14.24%	**0.5460**†	12.87%

Table 1: Comparisons between NPRF and *BM25* on *TREC1-3* dataset. Relative performances compared with *BM25* are in percentages. Significant improvements relative to the baselines are marked with †.

	Title						Description					
Model	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
BM25	0.2533	-	0.3612	-	0.4158	-	0.2479	-	0.3514	-	0.4110	-
NPRF$_{ff}$-DRMM	0.2823†	11.46%	0.3941†	9.11%	0.4350†	4.62%	0.2766†	11.58%	0.3908†	11.21%	0.4421†	7.56%
NPRF$_{ff'}$-DRMM	0.2837†	12.00%	0.3928†	8.74%	0.4377†	5.27%	0.2774†	11.90%	0.3984†	13.38%	0.4493†	9.32%
NPRF$_{ds}$-DRMM	**0.2904**†	14.66%	**0.4064**†	12.52%	**0.4502**†	8.28%	**0.2801**†	12.95%	**0.4026**†	14.57%	**0.4559**†	10.92%
NPRF$_{ff}$-KNRM	0.2809†	10.90%	0.3851†	6.62%	0.4287	3.11%	0.2720†	9.71%	0.3867†	10.06%	0.4356†	5.99%
NPRF$_{ff'}$-KNRM	0.2815†	11.13%	0.3882†	7.48%	0.4264	2.55%	0.2737†	10.39%	0.3892†	10.74%	0.4382†	6.61%
NPRF$_{ds}$-KNRM	0.2846†	12.36%	0.3926†	8.69%	0.4327	4.06%	0.2800†	12.95%	0.3972†	13.03%	0.4477†	8.94%

Table 2: Comparisons between NPRF and *BM25* on the *Robust04* dataset. Relative performances compared with *BM25* are in percentages. Significant improvements relative to the baselines are marked with †.

NIRM(QE) in most cases, indicating the benefit brought by wrapping up the feedback information in a document-to-document matching framework as in NPRF, as opposed to directly adding unweighted expansion terms to the query. Recall that, it is not straightforward to incorporate these expanded terms within the existing NIRMs' architectures because the NIRMs do not distinguish between them and the original query terms.

3.3 Analysis

Parameter sensitivity. Moreover, we analyze factors that may influence NPRF's performance. We report results on NPRF$_{ds}$ using title queries on Robust04 for the sake of brevity, but similar observations also hold for the other NPRF variants, as well as on TREC1-3. Figure 2 illustrates the sensitivity of NPRF relative to two parameters: the number of feedback documents m within D_q and the number of terms k that are used to summarize each $d_q \in D_q$. Specifically, Figure 2 shows the performance of NPRF$_{ds}$ as the number of feedback documents m varies (top), and as the number of top terms k varies (bottom). The effectiveness of NPRF appears to be stable over a wide range of the parameter configurations, where the proposed model consistently outperforms the BM25 baseline.

Case study. A major advantage of the proposed NPRF over existing neural IR models is that it allows for soft-matching query-related terms that are missing from both the query and the target document. Table 6 presents an illustrative example of soft matching in NPRF. From Table 6, it can be seen that there exist query-related terms in the top-10 documents returned by BM25 in the initial ranking. However, since those query-related terms are missing in both the query and the target document, they are not considered in the document-to-query matching and, consequently, the target document is ranked 122nd by BM25 despite the facts that it was judged relevant by a human assessor. In contrast, the NPRF framework allows for the soft-matching of terms that are missing in both the query and target document. As a result, the matching signals for the query terms and query-related terms in the target document are enhanced. This leads to enhanced effectiveness with the target document now ranked in the 5th position.

In summary, the evaluation on two standard TREC test collections shows promising results obtained by our proposed NPRF approach, which outperforms state-of-the-art neural IR models in most cases. Overall, NPRF provides effective retrieval performance that is robust with respect to the two embedded neural models used for encoding the document-to-document interactions, the two kinds of queries with varied length, and wide range of parameter configurations.

	Title						Description					
Model	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
DRMM	0.2469	-	0.4833	-	0.4919	-	0.2111	-	0.4423	-	0.4546	-
K-NRM	0.2284	-	0.4410	-	0.4530	-	0.1763	-	0.3753	-	0.3854	-
PACRR-firstk	0.2393	-	0.4620	-	0.4782	-	0.1702	-	0.3577	-	0.3666	-
NPRF$_{ff}$-DRMM	0.2669†	8.12%	0.5010	3.66%	0.5119	4.06%	0.2509†	18.83%	0.5257†	18.84%	0.5393†	18.63%
NPRF$_{ff'}$-DRMM	0.2671†	8.19%	0.5023	3.94%	0.5116	4.01%	0.2504†	18.61%	0.5163†	16.73%	0.5291†	16.40%
NPRF$_{ds}$-DRMM	0.2698†	9.26%	0.5187†	7.32%	0.5282†	7.38%	**0.2527**†	19.69%	**0.5283**†	19.44%	0.5444†	19.76%
NPRF$_{ff}$-KNRM	0.2633†	15.28%	0.5033†	14.13%	0.5171†	14.14%	0.2486†	40.97%	0.5240†	39.61%	0.5398†	40.06%
NPRF$_{ff'}$-KNRM	0.2654†	16.20%	0.5077†	15.12%	0.5216†	15.15%	0.2462†	39.65%	0.5197†	38.45%	0.5363†	39.13%
NPRF$_{ds}$-KNRM	**0.2707**†	18.51%	**0.5303**†	20.26%	**0.5406**†	19.35%	0.2505†	42.04%	0.5270†	40.41%	**0.5460**†	41.67%

Table 3: Comparisons between NPRF and neural IR models on *TREC1-3*. Relative performances of NPRF-DRMM(KNRM) compared with DRMM (K-NRM) are in percentages, and statistically significant improvements are marked with †.

	Title						Description					
Model	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
DRMM	0.2688	-	0.3713	-	0.4297	-	0.2630	-	0.3558	-	0.4135	-
K-NRM	0.2464	-	0.3510	-	0.3989	-	0.1687	-	0.2301	-	0.2641	-
PACRR-firstk	0.2540	-	0.3631	-	0.4082	-	0.2087	-	0.2962	-	0.3362	-
NPRF$_{ff}$-DRMM	0.2823	5.03%	0.3941†	6.14%	0.4350	1.24%	0.2766†	5.17%	0.3908†	9.84%	0.4421†	6.92%
NPRF$_{ff'}$-DRMM	0.2837†	5.55%	0.3928	5.78%	0.4377	1.87%	0.2774†	5.48%	0.3984†	11.97%	0.4493†	8.67%
NPRF$_{ds}$-DRMM	**0.2904**†	8.05%	**0.4064**†	9.46%	**0.4502**	4.78%	**0.2801**†	6.46%	**0.4026**†	13.15%	**0.4559**†	10.26%
NPRF$_{ff}$-KNRM	0.2809†	14.00%	0.3851†	9.72%	0.4287†	7.48%	0.2720†	61.22%	0.3867†	68.08%	0.4356†	64.96%
NPRF$_{ff'}$-KNRM	0.2815†	14.25%	0.3882†	10.60%	0.4264†	6.90%	0.2737†	62.21%	0.3892†	69.12%	0.4382†	65.93%
NPRF$_{ds}$-KNRM	0.2846†	15.50%	0.3926†	11.85%	0.4327†	8.47%	0.2800†	65.98%	0.3972†	72.62%	0.4477†	69.55%

Table 4: Comparisons between NPRF and neural IR models on *Robust04*. Relative performances of NPRF-DRMM(KNRM) compared with DRMM (K-NRM) are in percentages, and statistically significant improvements are marked with †.

4 Related Work

Recently, several neural IR models (NIRMs) have been proposed to apply deep learning techniques in ad-hoc information retrieval. One of the essential ideas from prior work is to model the document-to-query *interaction* via neural networks, based on a matrix of document-to-query embedding term similarities, incorporating both the “exact matching” of terms appearing in both the document and query and the “soft matching” of different query and document term pairs that are semantically related.

DSSM, one of the earliest NIRMs proposed in (Huang et al., 2013), employs a multi-layer neural network to project queries and document into a common semantic space. The cosine similarity between a query and a document (document title) is used to produce a final relevance score for the query-document pair. CDSSM is a convolutional version of DSSM, which uses the convolutional neural network (CNN) and max-pooling strategy to extract semantic matching features at the sentence level (Shen et al., 2014). (Pang et al., 2016) also employ a CNN to construct the MatchPyramid model, which learns hierarchical matching patterns between local interactions of document-query pair. (Guo et al., 2016) argue that both DSSM and CDSSM are representation-focused models, and thus are better suited to capturing semantic matching than relevance matching (i.e., lexical matching), and propose the interaction-focused relevance model named DRMM. DRMM maps the local interactions between a query-document pair into a fixed-length histogram, from which the exact matching signals are distinguished from the other matching signals. These signals are fed into a feed forward network and a term gating network to produce global relevance scores. Similar to DRMM, K-NRM (Xiong et al., 2017) builds its model on top of a matrix of local interaction signals, and utilizes multiple Gaussian kernels to obtain multi-level exact/soft matching features that are input into a ranking layer to produce the final ranking score. K-NRM is later improved by Conv-KNRM, which employs CNN filters to capture n-gram representations of queries and documents (Dai et al., 2018). DeepRank (Pang et al., 2017) models the relevance generation process by identifying query-centric contexts, processing them with a CNN or LSTM, and aggregating them to produce a final relevance score. Building upon DeepRank, (Fan et al., 2018) propose to model diverse relevance patterns by a data-driven method to allow rele-

	TREC1-3						Robust04					
	Title			Description			Title			Description		
Model	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20
BM25+QE	**0.2873**	0.5200	0.5330	**0.2601**	0.4973	0.5093	**0.2966**	0.3839	0.4353	**0.2926**	0.3817	0.4340
QL+RM3	0.2734	0.5093	0.5198	0.2421	0.4627	0.4801	0.2842	0.3878	0.4398	0.2686	0.3506	0.4150
DRMM (QE)	0.2741	0.5183	0.5345	0.2380	0.5077	0.5229	0.2876	0.4002	**0.4549**	0.2711	0.3822	0.4392
K-NRM (QE)	0.2633	0.5127	0.5235	0.2307	0.4877	0.5039	0.2521	0.3644	0.4062	0.2380	0.3304	0.3785
NPRF$_{ds}$-DRMM	0.2698	0.5187	0.5282	0.2527	**0.5283**	0.5444†	0.2904	**0.4064**	0.4502	0.2801	**0.4026**†	**0.4559**†
NPRF$_{ds}$-KNRM	0.2707	**0.5303**	**0.5406**	0.2505	0.5270	**0.5460**†	0.2846	0.3926	0.4327	0.2800	0.3972†	0.4477

Table 5: Comparisons between NPRF and *query expansion baselines* on *TREC1-3* and *Robust04*. Significant improvements over the best baseline is marked with †.

TREC Query 341: airport security	
Terms in doc at rank *i*	Terms in target document FBIS3-23332
1. terrorist detect passenger check police scan; 2. heathrow terrorist armed aviation police; 3. detect airline passenger police scan flight weapon; 4. aviation; 5. detect baggage passenger; 6. passenger bomb baggage terrorist explosive aviation scan flight weapon; 7. baggage airline detect passenger scan flight weapon; 8. baggage airline passenger flight; 9. passenger police aviation; 10. airline baggage aviation flight	transec semtex airline ditma **security** baggage heathrow test device lockerbie klm bomb virgin **airport** loaded blobby transport detect inspector terrorist identify atlantic depressing passenger fail aircraft dummy check inert patchy stein norwich doll regard rupert lapse busiest loophole employee campaign blew procedure traveler passport reconcile glasgow investigate boeing bags bag harry successive smuggle conscious reconciliation tragedy board wire hidden...

Table 6: An illustrative example of soft matching in NPRF. The target document FBIS3-23332, judged relevant, is ranked 122^{nd} by BM25 for query 341 on Robust04, and is promoted to the 5^{th} by NPRF$_{ds}$-DRMM. The NPRF mechanism increases the chances of soft-matching query-related terms that appear in the top-ranked documents (terms in blue), but are missing in both the query and the target document. Subsequently, the matching signals with the query terms (in **bold**) and the query-related terms (in red) in the target document are enhanced.

vance signals at different granularities to compete with each other for the final relevance assessment.

Duet (Mitra et al., 2017) employs two separate deep neural networks to build a relevance ranking model, in which a local model estimates the relevance score according to exact matches between query and document terms, and a distributed model estimates relevance by learning dense lower-dimensional representations of query and document text. (Zamani et al., 2018) extends the Duet model by considering different fields within a document.

(Hui et al., 2017) propose the PACRR model based on the idea that an appropriate combination of convolutional kernels and pooling operations can be used to successfully identify both unigram and n-gram query matches. PACRR is later improved upon by Co-PACRR, a context-aware variant that takes the local and global context of matching signals into account through the use of three new components (Hui et al., 2018).

(Ran et al., 2017) propose a document-based neural relevance model that utilizes complemented medical records to address the mismatch problem in clinical decision support. (Nogueira and Cho, 2017) propose a reinforcement learning approach to reformulating a task-specific query. (Li et al., 2018) propose DAZER, a CNN-based neural model upon interactions between seed words and words in a document for zero-shot document filtering with adversarial learning. (Ai et al., 2018) propose to refine document ranking by learning a deep listwise context model.

In summary, most existing neural IR models are based on query-document interaction signals and do not provide a mechanism for incorporating relevance feedback information. This work proposes an approach for incorporating relevance feedback information by embedding neural IR models within a neural pseudo relevance feedback framework, where the models consume feedback information via document-to-document interactions.

5 Conclusions

In this work we proposed a neural pseudo relevance feedback framework (NPRF) for incorporating relevance feedback information into existing neural IR models (NIRM). The NPRF framework uses feedback documents to better estimate relevance scores by considering individual feedback documents as different interpretations of the user's information need. On two standard TREC datasets, NPRF significantly improves the performance of two state-of-the-art NIRMs. Furthermore, NPRF was able to improve their perfor-

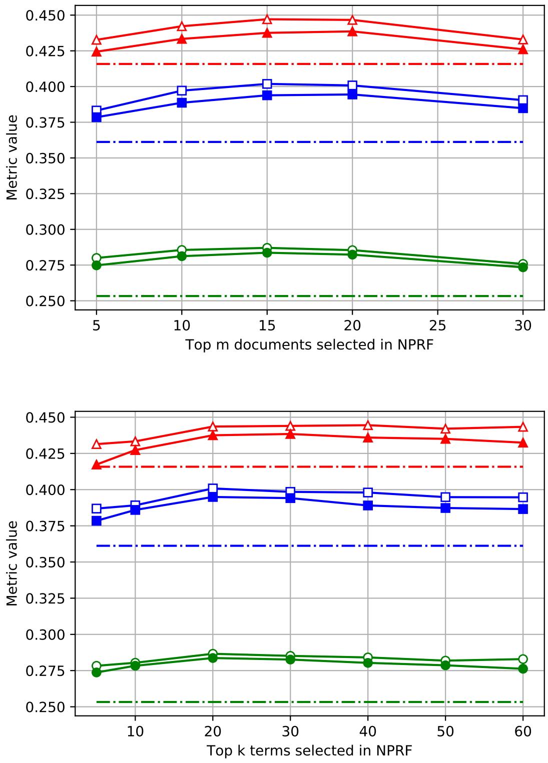

Figure 2: Performance of NPRF$_{ds}$ with different numbers of PRF documents (top) and different umber of terms which are used to summarize the feedback documents (bottom). The ○, □, △ correspond to results measured by MAP, P@20 and NDCG@20 respectively, and the empty or solid symbols correspond to those for NPRF$_{ds}$-DRMM and NPRF$_{ds}$-KNRM. The three dotted lines, from bottom to top, are the BM25 baseline evaluated by MAP, P@20 and NDCG@20, respectively.

mance across two kinds of query tested (namely, short queries and the verbal queries in natural language). Finally, our analysis demonstrated the robustness of the NPRF framework over different parameter configurations.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (61433015/61472391), and the Beijing Natural Science Foundation under Grant No. (4162067/4142050).

References

Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *SIGIR*, pages 135–144. ACM.

Chris Buckley and Stephen E. Robertson. 2008. Relevance feedback track overview: Trec 2008. In *TREC*. NIST.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*, pages 126–134. ACM.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *ACL*. The Association for Computer Linguistics.

Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling diverse relevance patterns in ad-hoc retrieval. In *SIGIR*, pages 375–384. ACM.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*, pages 55–64. ACM.

Donna Harman. 1993. Overview of the first text retrieval conference. In *SIGIR*, pages 36–47. ACM.

Donna Harman. 1994. Overview of the third text retrieval conference (TREC-3). In *TREC*, volume Special Publication 500-225, pages 1–20. National Institute of Standards and Technology (NIST).

Donna Harman. 1995. Overview of the second text retrieval conference (TREC-2). *Inf. Process. Manage.*, 31(3):271–289.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338. ACM.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A position-aware neural IR model for relevance matching. In *EMNLP*, pages 1049–1058. Association for Computational Linguistics.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2018. Co-pacrr: A context-aware neural IR model for ad-hoc retrieval. In *WSDM*, pages 279–287. ACM.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *SIGIR*, pages 120–127. ACM.

Chenliang Li, Wei Zhou, Feng Ji, Yu Duan, and Haiqing Chen. 2018. A deep relevance model for zero-shot document filtering. In *ACL*, pages 2300–2310. The Association for Computer Linguistics.

Craig Macdonald, Richard McCreadie, Rodrygo Santos, and Iadh Ounis. 2012. From puppy to maturity: Experiences in developing terrier. In *OSIR at SIGIR*, pages 60–63.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*, pages 1291–1299. ACM.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *EMNLP*, pages 574–583. Association for Computational Linguistics.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A study of matchpyramid models on ad-hoc retrieval. *CoRR*, abs/1606.04648.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *CIKM*, pages 257–266. ACM.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281. ACM.

Yanhua Ran, Ben He, Kai Hui, Jungang Xu, and Le Sun. 2017. A document-based neural relevance model for effective clinical decision support. In *BIBM*, pages 798–804. IEEE Computer Society.

Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Mike Gatford, and A. Payne. 1995. Okapi at TREC-4. In *TREC*, volume Special Publication 500-236. National Institute of Standards and Technology (NIST).

J. Rocchio. 1971. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART retrieval system: experiments in automatic document processing*, pages 313–323. Prentice Hall, Englewood, Cliffs, New Jersey.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *WWW (Companion Volume)*, pages 373–374. ACM.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 robust track. In *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST).

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*, pages 55–64. ACM.

Zheng Ye, Xiangji Huang, Ben He, and Hongfei Lin. 2009. York university at TREC 2009: Relevance feedback track. In *TREC*, volume Special Publication 500-278. National Institute of Standards and Technology (NIST).

Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. 2018. Neural ranking models with multiple document fields. In *WSDM*, pages 700–708. ACM.