A Document-Based Neural Relevance Model for Effective Clinical Decision Support

Yanhua Ran*, Ben He*, Kai Hui†, Jungang Xu* and Le Sun‡
*School of Computer & Control Engineering, University of Chinese Academy of Sciences, Beijing, China
Email: ranyanhua16@mails.ucas.ac.cn,{benhe, xujg}@ucas.ac.cn
†Max Planck Institute for Informatics, Saarbrücken Graduate School of Computer Science, Saarbrücken, Germany
Email: khui@mpi-inf.mpg.de
‡Institue of Software, Chinese Academy of Sciences, Beijing, China
Email: sunle@iscas.ac.cn

***Abstract*—Clinical Decision Support (CDS) can be regarded as an information retrieval (IR) task, where medical records are used to retrieve the full-text biomedical articles to satisfy the information needs from physicians, aiming at better medical solutions. Recent attempts have introduced the advances of deep learning by employing neural IR methods for CDS, where, however, only the document-query relationship is modeled, resulting in non-optimal results in that a medial record can barely reflect the information included in a relevant biomedical article which is usually much longer. Therefore, in addition to the document-query relationship, we propose a document-based neural relevance model (DNRM), addressing the mismatch by utilizing the content of relevant articles to complement the medical records. Specifically, our DNRM model evaluates a document relative to a query and to several pseudo relevant documents for the query at the same time, capturing the interactions from both parts with a feed forward network. Experimental results on the standard Text REtrieval Conference (TREC) CDS track dataset confirm the superior performance of the proposed DNRM model.**

I. Introduction

To make better clinical decisions, physicians often seek out for useful information from massive published biomedical literature for help. Given the huge volume of the publications and their rapid updates, it is necessary to develop systems to assist physicians in retrieving the biomedical literature more effectively. To achieve this, Clinical Decision Support (CDS) systems aim at bridging the electronic health records to the biomedical literature for patient care [1].

In general, CDS is regarded as an information retrieval (IR) task, in which the queries are patient records and the documents are biomedical articles. Most of the existing approaches are based on the traditional retrieval models which rely on the exact match between the terms from the health records and from the biomedical articles [1][2]. Such approaches have been demonstrated effective in Text REtrieval Conference (TREC) CDS tracks [1][2]. More recently, since deep learning promises enormous improvements in IR [4][5], neural IR models based on Word2Vec and the deep neural network (DNN) are also introduced to solve CDS tasks [3], where a D2Q (document-to-query) approach is developed to encode the cosine similarities between the embeddings of patient records and the biomedical articles. Intuitively, the usage of the embeddings incorporates the latent semantic information of context, enabling to model the matching signals beyond exact match, leading to improvements over the classical retrieval models [3][5]. However, the effectiveness of the neural approaches might be limited due to the heterogeneous comparisons where a query is usually much shorter than a document, resulting in the mismatch even when they are highly relevant, which, unfortunately, is especially serious in CDS tasks. In other words, it might be always the case that a patient record may fail to cover different aspects from a relevant biomedical article due to their difference, making a model solely depending on the document-query relationship underperformed.

To close this gap, in this paper, a novel document-based neural relevance model (DNRM) is proposed to mitigate the mismatch by incorporating the comparisons among different articles. Put differently, as a complement to the document-query matching, a query is expanded with several pseudo relevant documents to utilize the homogeneous matching among articles. In particular, given a patient record q and a biomedical article d, N pseudo relevant documents for q from an initial ranking are compared against d in terms of similarity, ending up with N similarity values, which are fed into a deep neural network to produce a scalar. Interpolated with the relevant score from the initial ranking, the produced ranking score thereby summarizes the matching between d and q with pseudo relevant documents as the intermediary. Apart from that, the document-query signals are also considered by summarizing the document-to-query similarity akin to the above methods, ending up with a relevant score directly based on d and q. Ultimately, the two relevant scores are combined as the final relevant score. Both parts and the combination are encoded in a unified end-to-end neural network, enabling the model in trading-off between the relevant signals from both parts. In the end, the experimental results on the standard TREC CDS 2014 and 2015 (A) track confirm the superior performance of the proposed DNRM model over CDS tasks.

The remainder of this paper is organized as follows. We recap the related works and put our work into context in Section II. Thereafter, in Section III, we describe the proposed DNRM model. Experimental settings and evaluation results are summarized in Section IV before we conclude this work in Section V.

II. RELATED WORK

A. Conventional Retrieval Approaches and BM25

Information retrieval aims to retrieve the documents relevant to a given user query. In general, the retrieved documents are ranked by the degree of relevance to the query, which is often measured by the scores given by IR models, such as the classical BM25 model [6]. Most traditional IR models are based on exact match, namely the count of the query terms in the document. Different IR models have different weighting and normalization schemes over these counts.

As one of the state-of-the-art traditional IR models, we utilize BM25 to produce the initial run in our approach. In addition, pseudo relevance feedback (PRF) is a popular method for improving IR effectiveness by using the top-ranked documents as pseudo relevance set, from which the most informative terms are expanded to the original query. One of the best-performing PRF methods on top of BM25 is an adaption of Rocchio's algorithm presented in [7], which is able to provide state-of-the-art retrieval effectiveness on standard TREC test collections [7].

B. Neural IR Models

In recent years, word embeddings and neural networks have been successfully applied in many NLP tasks. While word embeddings learned by neural network approaches, Word2Vec [9] for example, have already been explored in IR tasks [10][11], neural networks' influence on IR tasks is still remained to be explored and has attracted researchers' attentions. DSSM [12], C-DSSM [13] and CLSM [14] utilize click-through data to train a deep neural network to map queries and documents into embedding representations and then the relevance score for a given query-document pair is measured by the cosine distance between their embedding representations. These three models use titles of documents for retrieval, which are much shorter than the full text of the documents. Instead of learning embedding representations for queries and documents through neural network, MatchPyramid [15] is based on interaction signals between terms from two different pieces of text to be matched. It first constructs a term-to-term matching matrix, whose values are similarities (like cosine similarities of their embeddings) of the corresponding two terms. Then a deep neural network composed by convolutional neural network (CNN) layers followed by an MLP is applied on the matching matrix to obtain the final matching score for the given two pieces of text. A later study of MatchPyramid models on ad-hoc retrieval indicates that MatchPyramid can obtain close results with traditional models, such as QLM and BM25, but still worse than traditional models [4]. Guo et al. [5] propose a novel relevance matching model named DRMM using deep neural network for ad-hoc retrieval. DRMM is also an interaction-based model which is based on the count of similarities between query terms and document terms. It computes a score for every query term of a given query through a feed forward network and a term gating network is used to fuse the scores to obtain the final matching score.

C. State-of-the-art CDS Methods

Palotti & Hanbury first utilize MetaMap to map original query to Unified Medical Language System (UMLS) concepts. After that, some relevant concepts are selected to expand original query with different weights. Pseudo-relevance feedback is also applied to further expand the query [16]. Song et al. expand original query by adding the relevant MeSH terms of the titles and snippets retrieved by Google [17]. In addition to query expansion, learning-to-rank algorithms based on pointwise or pairwise and a query term position-based approach are applied to re-rank the initial results [17]. Considering that the documents in CDS task are very long, Cummins et al. apply a document language model SPUD [18] to improve the retrieval performance [19]. Abacha & Khelifi explore several query expansion methods utilizing MeSH and DBpedia terms and several result fusion approaches, such as rank-based and score-based approaches, are explored to improve performance [20]. Choi & Choi incorporate the query types (diagnosis, test and treatment) information by training a classifier to rank documents and then the topic-specific ranking score is fused with the relevance score given by query likelihood model (QLM) with query expansion [21]. Balaneshin-kordan et al. extract unigrams, bigrams and multi-word UMLS concepts from different sources, such as queries, pseudo-relevance feedback documents and external knowledge resources, and then use the Markov Random Field (MRF) model to rank documents [23] .

Gurulingappa et al. [25] propose a semi-supervised method that takes the advantages of pseudo-relevance feedback, semantic query expansion and document similarity measures based on unsupervised word embeddings. Firstly, terms expanded by the UMLS concepts and document titles in the top-k pseudo relevance feedback set are extracted and added with a weight of 0.1 to the initial query. Secondly, by using the unsupervised word embedding method, centroids of articles are computed based on the abstract, the title or the journal title. Finally, ranking scores obtained from PRF, UMLS expansion and word embedding document distances are used as features in the supervised learning to rank model.

Yang & He explore to integrate the semantic similarity between embeddings of the patient record and biomedical article to improve the performance of CDS system [3]. The semantic similarity score ultimately is interpolated with the BM25 baseline model to obtain the final score, which is utilized to re-rank the baseline results.

III. DOCUMENT-BASED NEURAL RELEVANCE MODEL

In this section, we describe the proposed DNRM which is summarized in Figure 1. Given a document-query pair as well as several pseudo relevant documents via relevant feedback, there are two components in parallel: the component which consumes the similarity between a document and the individual terms from a query (*D2Q*), and the component which digests the similarity between a document and all documents from the relevance feedback (*D2D*). In both components, features are first extracted in terms of a variant of cosine

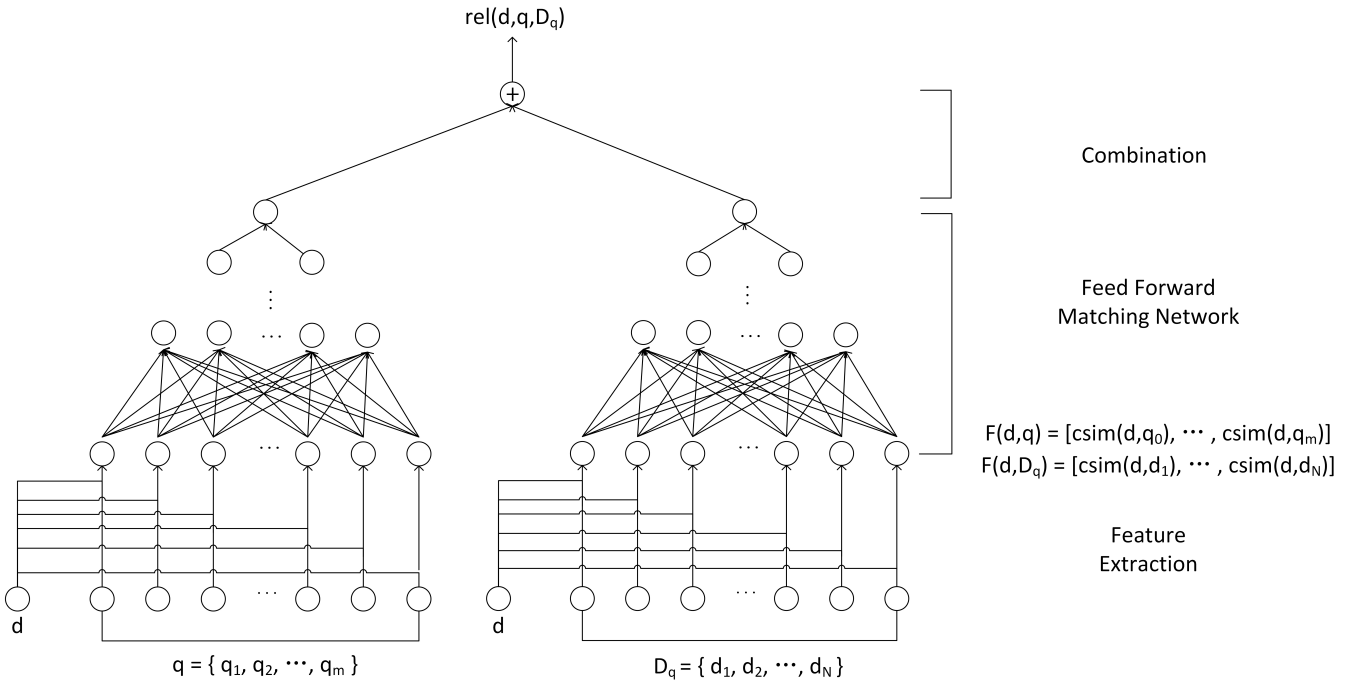


Fig. 1. The architecture of the proposed DNRM model. d is a document to be evaluated relative a query q. There are two components in parallel: the document-to-query (D2Q) in the left and the document-to-document (D2D) in the right. Scores from both parts are combined to generate the ultimate relevance score.

similarity, and subsequently several dense layers are employed to summarize these signals with a scalar. Ultimately, another dense layer is utilized to combine these two scalars from either component and generate the relevance score.

A. Notation

A query q with m terms and a document d_i with n terms are both represented as a sequence of terms, namely, $[q_1, q_2, \cdots, q_m]$ and $[d_{i1}, d_{i2}, \cdots, d_{in}]$. In addition to a document d, in the D2D component, top-N documents from an initial search result for q serve as the pseudo relevant documents, which are denoted as $D_q = \{d_1, d_2, ..., d_N\}$, where N is the number of documents. To compute the similarity between a term and a document or between two documents, one needs to first embed them into a vector space. Thus, we denote $w2v(.)$ and $d2v(.)$ as two embedding functions for a term and a document respectively. A variant of cosine similarity, denoted as $csim(.,.)$, is employed in this work, based on which features $F(d,q)$ and $F(d,D_q)$ are extracted for D2Q and D2D. The outcome scores from D2Q and D2D, as well as the ultimate score, are denoted as $rel_q(d_i,q)$, $rel_{D_q}(d_i,D_q)$ and $rel(d_i,q,D_q)$ respectively.

B. Model Architecture

Given the word embedding of all terms, we introduce the D2Q and D2D components in this section.

Embed documents into vector space. When embedding a document into a vector space, one desires to preserve the semantic meaning of the document meanwhile employing fewer dimensions for the sake of efficiency. Different from the methods used by Yang & He [3], in this work, we embed a document d_i by averaging the embeddings of all its terms, weighted by the *tfidf* scores of individual terms. The formula is summarized in Equation (1). The $tfidf(t)$ represented the *tfidf* score of a term t as indicated in the following, where N_d is the total number of the documents in the collection, tf(t) and df(t) denote the term frequency and the document frequency of a term t respectively.

$$tfidf(t) = tf(t) log_2 \frac{N_d + 0.5}{df(t) + 0.5}$$

Intuitively, a well-trained word embedding should preserve the semantic meaning of a term [9]. When employing weighted average of word embeddings in d, the terms contribute most of the semantic meaning of d may also determine the derived embedding most, leading to a semantic embedding for d.

$$vec(d_i) = \sum_{d_{ij} \in d_i} tfidf(d_{ij}) \vec{d_{ij}} \quad (1)$$

The similarity function adopted in this paper, namely $csim(.,.)$, is a variant of cosine similarity as in Equation (2).

$$csim(.,.) = 0.5 * cosine(.,.) + 0.5 \quad (2)$$

D2Q. Given a document d and a query q, we first compute the similarity between d and individual query terms, resulting in m similarities for a document-query pair, which is summarized in the following equation.

$$F(d,q) = [csim(d,q_0), csim(d,q_1), \cdots, csim(d,q_m)]$$

Intuitively, this setting particularly caters for the facts that the medical records are usually quite long in CDS and one would desire to preserve the matching signals for all different query terms in follow-up process. This actually enables the model to take the query coverage into consideration by rewarding the documents that can more comprehensively cover different aspects in a query. Subsequently, l dense layers are employed to learn from $F(d, q)$, ending up with a scalar $rel_q(d, q)$ which represents the relevance between d and q. The dense layers are summarized in the following.

$$
\begin{aligned}
{h_q}^{(0)} &= F(d, q) \\
{h_q}^{(1)} &= tanh(W_q^{(1)} {h_q}^{(0)} + {b_q}^{(1)}) \\
&\cdots \\
{h_q}^{(l)} &= tanh(W_q^{(l)} {h_q}^{(l\text{-}1)} + {b_q}^{(l)})
\end{aligned}
$$

Intuitively, the dense layers not only consider how d satisfies the information need represented by individual query terms q_i, but also take their interactions into consideration, making the model fully aware of the relationship between q and d.

D2D captures the semantic relationship between a document d and different pseudo relevant documents for a query q. Akin to D2Q, the similarity is computed between d and individual documents, namely, all $d_i \in D_q$, leading to a list of similarity signals which are summarized in the following.

$$F(d, D_q) = [csim(d, d_1), csim(d, d_2), \cdots, csim(d, d_N)]$$

Subsequently, several dense layers are employed to compute $rel_{D_q}(d, D_q)$. Though dense layers are employed in both D2Q and D2D, we argue that they are set up for different purposes. Namely, when evaluating a document d, documents from D_q actually serve as different prototypes, attempting to interpret q from different perspectives in the space of d, mitigating the gaps between q and d. Beyond that, the ranking of documents are also employed by the dense layers as an indicator about the confidence for a particular document.

Combination of the signals from D2Q and D2D. A dense layer is employed to combine the relevance scores from D2Q ($rel_q(d, q)$) and D2D ($rel_{D_q}(d, D_q)$). Intuitively, $rel_q(d, q)$ directly evaluates based on d and q which is shared by established relevance weighting models. Meanwhile, as a complement, $rel_{D_q}(d, D_q)$ measures the relevance via several pseudo relevant documents, where the information need is further interpreted in details by different top-ranked documents. In the combination, the model is learned to weight these two signals and produce an ultimate score $rel(d, q, D_q)$.

C. Loss Function and Model Training

Given a query q, a relevant document d^+, and a non-relevant document d^-, a widely-used max-margin loss is employed for training as in Equation (3).

$$
\begin{aligned}
&\mathcal{L}(q, D_q, d^+, d^-) \\
=&max(0, 1 - rel(d^+, q, D_q) + rel(d^-, q, D_q))
\end{aligned}
\tag{3}
$$

In the training time, given a query q and the ground-truth judgments, in each iteration, a relevant document is first sampled, associating to which, five to ten (a random number) non-relevant documents are sampled to pair with the relevant document, ending up with five to ten triples for training, e.g., (q, d^+, d^-), in the hope that the training data is fed to the model stochastically. In this work, only binary judgments are considered in the training. Adam [27] is employed for optimization where batch size is set to 16 to fit our model into the main memory.

IV. Experiment

In this section, we conduct experiments to compare the proposed DNRM with multiple the state-of-the-art methods from both unsupervised retrieval models and neural IR models.

TABLE I
THE RESULTS OBTAINED ON THE *Summary* FIELD OF THE TREC CDS 2014 TASK.

Method	infNDCG	infAP	R-prec	P@10	MAP
$BM25$	0.2640	0.0763	0.2264	0.3833	0.1668
$SEM\text{-}QD$	0.2688	0.0786	0.2318	0.3867	0.1719
$DRMM$	0.2666	0.0770	0.2270	0.3867	0.1675
$DNRM_Q$	0.1700	0.0327	0.1206	0.1700	0.0839
$DNRM_D$	0.2783	0.0799	0.2052	0.2733	0.1415
$DNRM_{D\text{-}\lambda}$	**0.3055***	**0.0875**	**0.2409**	**0.4200**	**0.1803**
$DNRM_{DQ\text{-}\lambda}$	0.2903	0.0844	0.2319	0.4067	0.1739

A. Experimental Setting

Dataset. The experiments are based on the standard test collection from TREC CDS Track 2014 and 2015 (A), which includes a snapshot of the open access subset of PubMed Central (PMC)[1] on January 21, 2014. There are 733,138 full-text biomedical articles of NXML format (XML encoded using the NLM Journal Archiving and Interchange Tag Library) in total. The *title*, *abstract*, *keywords* and *body* fields are extracted for indexing after stemming with Porter's stemmer and the removal of stopword. On average a document includes 2,583 tokens. All experiments are conducted with Terrier [29].

Topics. The number of the topics of TREC CDS 2014 and 2015 (A) are both 30. These topics are medical case narratives created by experts to serve as idealized representations of actual medical records. Each topic includes *summary* and *description* fields, where the description is a more comprehensive and verbose version of the topic. The results for both fields are reported.

Competing methods. As one of the state-of-the-art unsupervised models, BM25 with query expansion via Rocchio's PRF [7] is employed which is coined as $BM25$. Beyond that, as one of the state-of-the-art neural IR models, the DRMM model from Guo et al. [5] is implemented for comparison, coined as $DRMM$. We also investigate another neural IR model MatchPyramid [4], which, however, performs consistently worse than DRMM in our preliminary experiments and

[1] http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

is thereby omitted. In addition, we implement a semantic-based approach from Yang & He [3] which is coined as $SEM\text{-}QD$. In this work, DRMM is boosted by the interpolation with $BM25$. For DNRM model, the results for two variants are reported separately: the one only employs D2Q features, coined as $DNRM_Q$, and the one only employs D2D features, coined as $DNRM_D$. In addition, we also interpolate the relevance score from $BM25$ with different variants of $DNRM$, ending up with $DNRM_{D\text{-}\lambda}$, representing $DNRM_D$ with interpolation from $BM25$; $DNRM_{DQ\text{-}\lambda}$, representing $DNRM_{DQ}$ with interpolation from $BM25$, where $DNRM_{DQ}$ means both D2Q and D2D features are employed. Given a query, top-1,000 documents are first retrieved by BM25 with Rocchio's PRF. Thereafter, other baseline models re-rank these search results.

Cross validation. Due to the limit number of labeled data, all results are reported based on a five-fold cross validation. Models are tuned based on mean average precision (MAP) as in Guo et al. [5].

Metrics. We employ infNDCG of the top-ranked 1,000 documents as our primary metric, following the official setting in TREC CDS task, considering that inferred metric is more accurate when judging a relatively small number of documents and it distinguishes multiple levels (definitely relevant, possibly relevant, not relevant) of relevance. In addition, inferred Average Precision (infAP), R-Precision (R-Prec), P@10 and Mean Average Precision (MAP) are also reported for a comprehensive comparison among different methods. The latter four metrics only consider binary relevance. Significant difference is reported based on 95% confidence level (p_value < 0.05) from a two-tailed Student's t-test.

Hyper-parameters. The free parameters k_1 and k_3 of BM25 are set to $k_1 = 1.2$ and $k_3 = 1000$, following the default setting [6]. Grid search is applied to tune b, the number of feedback documents ED, and the number of expanded query terms ET. The size m in $F(d, q)$ is set to the maximum length of the queries. Zeros are padded to the tail of $F(d, q)$ when the length of a given query q is less than m. The size N in $F(d, D_q)$ is empirically set to 100 and 10 for TREC CDS 2014 and 2015 (A) respectively in our experiments. The number of layers for the D2Q component's and D2D component's dense layers are set to 3. The hidden layer sizes are set to the half of their input feature's size, which have little influence on the retrieval performance based on our pilot experiments. The matching network settings of DRMM follow the configuration in Guo et al. [5].

Training of Word2Vec. The word embeddings are trained based on a pool of top 1,000 documents returned by each of the queries as suggested in Diaz et al. [10]. The implementation of Word2Vec[2] from Mikolov et al. [9] is employed. In particular, we employ skip-gram, set the dimension to 100, the subsampling threshold to 10^{-3}, and we train for 20 iterations.

TABLE II
THE RESULTS OBTAINED ON THE *Description* FIELD OF THE TREC CDS 2014 TASK.

Method	infNDCG	infAP	R-prec	P@10	MAP
$BM25$	0.2402	0.0701	0.2104	0.3267	0.1501
$SEM\text{-}QD$	0.2450	0.0727	**0.2177**	**0.3533**	0.1577
$DRMM$	0.2418	0.0702	0.2106	0.3300	0.1502
$DNRM_Q$	0.1304	0.0234	0.1031	0.1233	0.0686
$DNRM_D$	0.2556	0.0712	0.1948	0.3300	0.1403
$DNRM_{D\text{-}\lambda}$	0.2493	0.0735	0.2146	0.3233	0.1552
$DNRM_{DQ\text{-}\lambda}$	**0.2558**	**0.0767**	0.2074	**0.3533**	**0.1578**

TABLE III
THE RESULTS OBTAINED ON THE *Summary* FIELD OF THE TREC CDS 2015 (A) TASK.

Method	infNDCG	infAP	R-prec	P@10	MAP
$BM25$	0.2841	0.0680	0.2269	0.4900	0.1713
$SEM\text{-}QD$	0.2860	0.0709	0.2281	0.4900	0.1741
$DRMM$	0.2875	0.0709	0.2316	0.4833	0.1746
$DNRM_Q$	0.1589	0.0265	0.1377	0.2000	0.0968
$DNRM_D$	0.2825	0.0743	0.2232	0.4100	0.1682
$DNRM_{D\text{-}\lambda}$	**0.3086***	**0.0814***	**0.2478***	**0.4933**	**0.1878***
$DNRM_{DQ\text{-}\lambda}$	0.3051	0.0788	0.2475*	0.4933	0.1869*

B. Results

The evaluation results for the summary and the description fields of TREC CDS 2014 and 2015 (A) are presented in Tables I - IV respectively, where the best results are highlighted in bold. Significant improvements over the best baseline are highlighted by a star. From Tables I - IV, it can be seen that: two variants with interpolation from DNRM, namely, $DNRM_{D\text{-}\lambda}$ and $DNRM_{DQ\text{-}\lambda}$, outperform all baselines on both summary and description in terms of infNDCG and infAP. Considering that they are not interpolated with $BM25$, the results of $DNRM_D$ are good enough to demonstrate the effectiveness of the proposed D2D feature. In particular, $DNRM_D$ obtains higher infNDCG, which is the primary metric to be considered, on both summary and description of TREC CDS 2014. They also achieve comparable infNDCG with all the baselines on summary of TREC CDS 2015 (A). It appears that $DNRM_Q$ does not perform as good as $DNRM_D$. Our explanation is that the computation of D2D similarity involves the use of term information from the whole documents, in which the query information is already included. Therefore, $DNRM_D$ outperforms $DNRM_Q$ in all cases. Obviously, $DNRM_{D\text{-}\lambda}$ outperforms $DNRM_{DQ\text{-}\lambda}$ on summary field. For description, $DNRM_{DQ\text{-}\lambda}$ obtains better results than $DNRM_{D\text{-}\lambda}$ on TREC CDS 2014 and obtains comparable results on TREC CDS 2015 (A). A possible reason is that description field is much longer than the summary and it can provide additional information which is not contained in D2D feature. As for the significance tests, it should be noted

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

that there are only 30 queries in both years, which makes it hard to obtain significant difference.

TABLE IV
THE RESULTS OBTAINED ON THE *Description* FIELD OF THE TREC CDS 2015 (A) TASK.

Method	infNDCG	infAP	R-prec	P@10	MAP
$BM25$	0.2683	0.0668	0.2125	0.4200	0.1586
$SEM\text{-}QD$	0.2729	0.0690	0.2187	0.4267	0.1640
$DRMM$	0.2727	0.0680	0.2127	0.4267	0.1611
$DNRM_Q$	0.1377	0.0241	0.1254	0.1700	0.0838
$DNRM_D$	0.2484	0.0699	0.2047	0.3233	0.1449
$DNRM_{D\text{-}\lambda}$	**0.2864**	0.0771	**0.2315***	0.4200	**0.1715**
$DNRM_{DQ\text{-}\lambda}$	0.2852	**0.0779**	0.2254	**0.4300**	0.1709

More analysis on DNRM. As mentioned in Section III, document embedding is obtained through the weighted average of the word embeddings in a document. Another choice is to choose the top-K terms according to their weights. The size N of the feedback document set D_q is also configurable. To have a better understanding of DNRM model, we further conduct the following experiments to study the impacts of the computation of document embedding and N. Firstly, we experiment with $K \in [0, 5, 10, ..., 100]$ (In particular, 0 represents the use of all terms in the document), using $N = 100$ and $N = 10$ for TREC CDS 2014 and 2015 (A) respectively. Moreover, we experiment with $N \in [5, 10, ..., 100]$ with $K = 0$. The results based on $DNRM_{D\text{-}\lambda}$ are displayed in Figures 2 and 3 in terms of infNDCG, where the horizontal line shows the results for $BM25$. From Figure 2 we can see that the performance of $DNRM_{D\text{-}\lambda}$ model varies in a small range without an obvious optimal peak with different K. A possible explanation is that the final document embedding is dominated by the terms with high weights. As for the different settings for N, the results, as shown in Figure 3, tend to have optimal values. The highest scores for summary and description witness a 20.3% and a 6.5% improvement relative to $BM25$ for TREC CDS 2014 respectively. Meanwhile, the best performance on summary and description are 8.6% and 9.5% higher than $BM25$ for TREC CDS 2015 (A). The results indicate that N should be carefully set when applying the DNRM model.

Comparison to the best-performed Run in TREC. In this section, we further take the best automatic and manual runs [23] from TREC CDS 2015 (A) task as the initial results in place of $BM25$ to investigate how DNRM model works with strong baselines. The evaluation results are summarized in Tables V and VI. From the results we can see even if the initial results are obtained by strong enough baseline models, the DNRM model can still improve on top of them. We argue that this is very important since a re-ranker is supposed to work well with all different baseline runs. More importantly, it should improve, at least not hurt, the performance when dealing with initial runs which have been ranked very well.

TABLE V
THE RESULTS OBTAINED ON THE *Summary* FIELD OF THE TREC CDS 2015 (A) TASK WHEN USING THE BEST *automatic* RUN AS THE INITIAL RUN.

Method	infNDCG	infAP
WSU-IR	0.2939	0.0842
DNRM	**0.3022***	**0.0864**

TABLE VI
THE RESULTS OBTAINED ON THE *Summary* FIELD OF THE TREC CDS 2015 (A) TASK WHEN USING THE BEST *manul* RUN AS THE INITIAL RUN.

Method	infNDCG	infAP
WSU-IR	0.3109	0.0880
DNRM	**0.3253***	**0.0917**

V. CONCLUSIONS AND FUTURE WORK

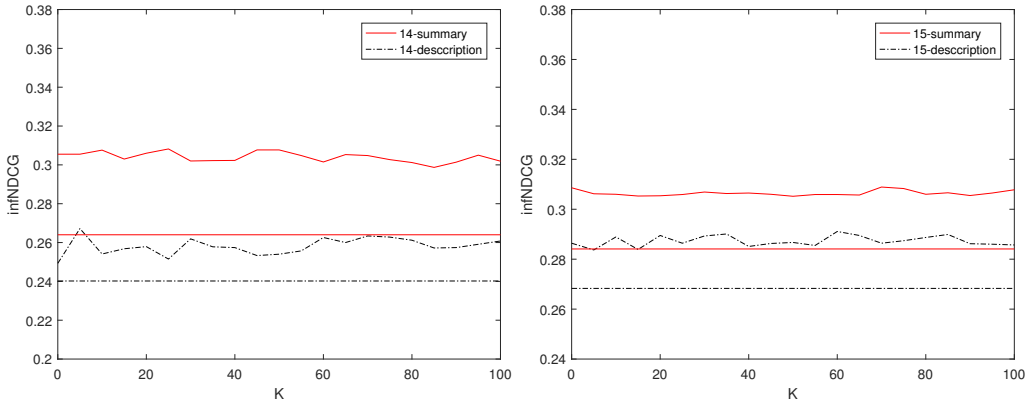
In this paper, we propose a novel document-based neural relevance model (DNRM) for Clinical Decision Support (CDS) task, in which the D2Q or D2D features are fed into several dense layers to obtain a neural relevance score. Experimental results indicate the proposed interpolation variants of DNRM model can outperform all the baselines in most cases. In fact, the results also indicate that by solely adopting D2D features one can already achieve good results. The D2Q feature is also an effective feature which is expected to enhance the performance of other IR models. In the future, we plan to incorporate field information as features input to the feed forward network, since they have been demonstrated to be useful [30]. We also plan to integrate the D2D features into the recently proposed PACRR model that learns k-gram features from a convolutional layer [8].

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (61472391/61433015/61572477).

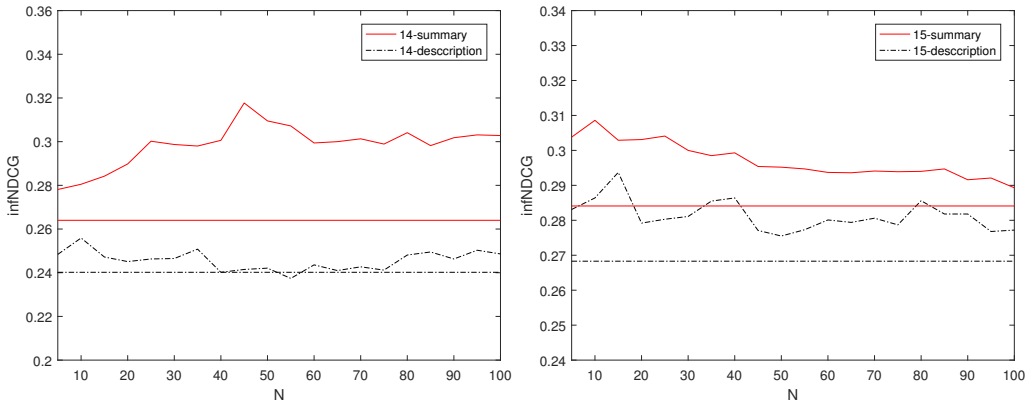
REFERENCES

[1] M. S. Simpson, E. M. Voorhees, and W. Hersh, "Overview of the TREC 2014 clinical decision support track." in *TREC*, 2014.
[2] K. Roberts, M. S. Simpson, E. M. Voorhees, and W. R. Hersh, "Overview of the TREC 2015 clinical decision support track." in *TREC*, 2015.
[3] C. Yang and B. He, "A novel semantics-based approach to medical literature search," in *IEEE* International Conference on Bioinformatics and Biomedicine, in *BIBM*, pp. 1616–1623, 2016.
[4] L. Pang, Y. Lan, J. Guo, J. Xu, and X. Cheng, "A study of Matchpyramid models on ad-hoc retrieval," *CoRR*, vol. abs/1606.04648, 2016.
[5] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *CIKM*, pp. 55–64, 2016.
[6] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," in *TREC*, 1995.
[7] K. Hui, B. He, T. Luo, and B. Wang, "A comparative study of pseudo relevance feedback for ad-hoc retrieval," in *ICTIR*, pp. 318–322, 2011.
[8] K. Hui, A. Yates, K. Berberich, and G. de Melo, "PACRR: A Position-Aware Neural IR Model for Relevance Matching," in *EMNLP*, pp. 1060–1069, 2017.



(a) 2014 (b) 2015

Fig. 2. Performance comparison of $DNRM_{D\text{-}\lambda}$ over different K.



(a) 2014 (b) 2015

Fig. 3. Performance comparison of $DNRM_{D\text{-}\lambda}$ over different N.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[10] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *ACL* 2016.

[11] D. Roy, D. Ganguly, M. Mitra, and G. J. F. Jones, "Word vector compositionality based relevance feedback using kernel density estimation," in *CIKM*, pp. 1281–1290, 2016.

[12] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. P. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *CIKM*, pp. 2333–2338, 2016.

[13] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, C. Chung, A. Z. Broder, K. Shim, and T. Suel, Eds. ACM, 2014, pp. 373–374.

[14] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *CIKM*, pp. 101–110, 2014.

[15] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *AAAI*, pp. 2793–2799, 2016.

[16] J. Palotti and A. Hanbury, "TUW@ TREC clinical decision support track 2015," Vienna University of Technology, Vienna Austria, Tech. Rep., 2015.

[17] Q. Hu, L. He, Y. Song, and Y. He, "ECNU at 2015 CDS track: Two re-ranking methods in medical information retrieval," in *TREC*, 2015.

[18] R. Cummins, J. H. Paik, and Y. Lv, "A pólya urn document language model for improved information retrieval," *ACM Trans. Inf. Syst.*, vol. 33, no. 4, pp. 21:1–21:34, 2015.

[19] R. Cummins, "Clinical decision support with the SPUD language model," in *TREC*, 2015.

[20] A. B. Abacha and S. Khelifi, "LIST at TREC 2015 clinical decision support track: Question analysis and unsupervised result fusion," in *TREC*, 2015.

[21] S. Choi and J. Choi, "SnuMedInfo at TREC CDS track 2014: Medical case-based retrieval task," in *TREC*, 2014.

[22] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *SIGIR*, pp. 275–281, 1998.

[23] S. Balaneshinkordan, A. Kotov, and R. Xisto, "WSU-IR at TREC 2015 clinical decision support track: Joint weighting of explicit and latent medical query concepts from diverse sources," in *TREC*, 2015.

[24] D. Metzler and W. B. Croft, "A Markov random field model for term dependencies," in *SIGIR*, pp. 472–479, 2005.

[25] H. Gurulingappa, L. Toldo, C. Schepers, A. Bauer, and G. Megaro, "Semi-supervised information retrieval system for clinical decision support," in *TREC*, 2016.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016,

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[28] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *NIPS*, pp. 402–408, 2000.

[29] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis, "From puppy to maturity: Experiences in developing terrier," in *OSIR at SIGIR*, pp. 60–63, 2012.

[30] S. Mohan, N. Fiorini, S. Kim, and Z. Lu, "Deep learning for biomedical information retrieval: Learning textual relevance from click logs," *BioNLP 2017*, pp. 222–231, 2017.