Neural Precision Medicine by Mining Implicit Treatment Concepts

Canjia Li[1,2] **Ben He**[1,2✉] **Le Sun**[2] **Yingfei Sun**[1✉]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing, China
licanjia17@mails.ucas.ac.cn {benhe, yfsun}@ucas.ac.cn sunle@iscas.ac.cn

***Abstract*—Precision Medicine (PM) is regarded as an information retrieval (IR) task, in which biomedical articles containing treatment information about specific diseases or genetic variants are retrieved in response to patient record, aiming at providing medical evidence to the point-of-care. In existing PM approaches, manual keywords such as "treatment" and "therapy" are considered direct indicators of treatment information, and are thereby introduced to expand the original query. However, the common medical concepts that are implicitly related to treatment (such as "oncogene", "tumor"), and differ the relevant documents from the non-relevant ones, are yet to be utilized. To bridge the gap, in this paper, we propose an extension of the state-of-the-art K-NRM neural retrieval model, coined K-NRM$^{\mathrm{PM}}$, to encapsulate the PM solutions within a neural network framework. Specifically, the proposed approach mines a global list of common medical concepts from documents that are judged pertinent to different queries. Thereafter, the mined implicit concepts are incorporated within a neural IR framework to enhance the effectiveness of precision medicine. Experimental results on the standard Text REtrieval Conference (TREC) PM track benchmark confirm the superior performance of the proposed K-NRM$^{\mathrm{PM}}$ model.**

I. Introduction

Precision Medicine (PM) systems aim at guiding precision oncologists to the best-known treatment options for the patient's condition. Recently, there has been increasing interest in PM in the field of medical search. For instance, the established Text REtriecal Conference (TREC) has been running a PM track[1] since 2017, aiming at providing a standard benchmark for PM experimentation [1]. As defined by the TREC PM track, a key feature in the PM biomedical article retrieval is that the query topics are mostly related to cancer, and a relevant document must be related to treatments. Neither diagnosis nor test about a disease is deemed relevant in that treatment is of paramount importance once a patient is diagnosed with cancer. As demonstrated in an example in Table I, a semi-structured PM query normally consists of four key elements: 1) disease names, e.g., type of cancer; 2) genetic variants, mostly in the tumors themselves instead of the patient's DNA; 3) demographic information, including age, gender, etc.; and 4) other factors that can possibly impact certain treatment options [1].

In response to a PM query, most existing approaches are composed of the following two components. The first component employs the classical ranking models, such as BM25 [2], to retrieve documents relevant to the disease and genetic variants, as specified in the query. The second component examines if the retrieved documents contain treatment information, and thereafter, adjusts the document ranking accordingly [1]. A popular choice of the second component is to manually create a keyword list that are supposed to be indicative of the treatment information. The keyword list serves as a boolean filter to drop documents that do not contain any of the words on the list [3][4].

TABLE I
An example of PM query topic taken from TREC 2017 PM Track.

Disease: Colon cancer
Gene: KRAS (G13D), BRAF (V600E)
Demographic: 52-year-old male
Other: Type II Diabetes, Hypertension

In addition to the direct indicators of treatment used in the manual keyword list, we argue that there exist terms and concepts about the treatment information that are in common across different PM queries. For instance, keywords like "mutation" and "tumor" are not explicit expression of treatment, but are implicitly related to articles about cancer treatment, and therefore can be used as complementary evidence of treatment information. To bridge the gap, in this paper, we propose a dictionary-based term generation method for the treatment aspect. In particular, the manually selected terms to account for treatment is regarded as explicit query to expand the original query. Moreover, the automatically learned common treatment concepts from the collection serve as a global implicit query for emphasizing on the treatment aspect. We extend the state-of-the-art neural retrieval model, K-NRM [5], by proposing the K-NRM^{PM} model that incorporates both explicit and implicit queries within a generic neural framework. Ultimately, the relevance scores from both queries are aggregated by a unified end-to-end neural network, which enables the model to trade-off between the relevance signals from both parts. The experiments on the standard TREC 2017 PM track benchmark confirm the superior performance of the proposed K-NRM^{PM} model. The contributions of this work are threefold: 1) the novel neural PM framework; 2) automatic implicit dictionary generation for the common treatment concepts; and 3) the experiments that confirm the effectiveness of the proposed neural framework.

[1]http://www.trec-cds.org/

978-1-5386-5488-0/18/$31.00 ©2018 IEEE

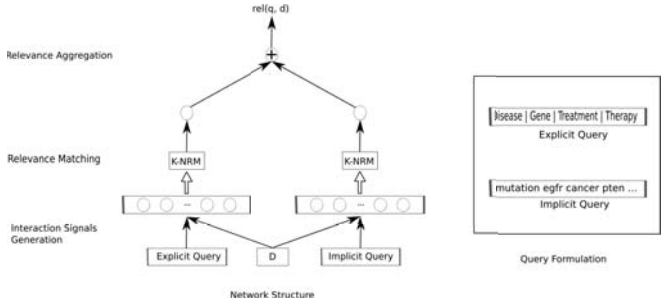


Fig. 1. Architecture of the proposed K-NRM^{PM} model. The model takes two separate queries as input. One is the explicit query composed of the original query and the manual treatment terms, and the other is the global implicit query mined from relevant documents of different training queries. The relevance matching module is instantiated by K-NRM [5], which encodes the relevance signals from both queries into scalars. The scalars are eventually aggregated through a dense layer to produce the final relevance score $rel(q, d)$ of a document d in response to query q.

II. Method

In this section, we present the proposed K-NRM^{PM} model which is summarized in Figure I. Given a document-query pair, K-NRM^{PM} estimates the relevance degree of the document from two separate perspectives: an explicit query expressing the local information need from user, concatenated with a list of terms that are related to explicit expression of treatment related information; and an implicit query formed by a dictionary automatically mined from the document collection itself, emphasizing on the implicit aspects of treatments. The K-NRM ranking model [5] is embedded to generate the relevance scores for both explicit and implicit queries. Ultimately, a dense layer is used to aggregate these scores from both components to produce the final relevance score.

A. Notation

A set of relevant documents is denoted as $D(rel)$ and documents in the whole collection as $D(all)$. $D(rel)$ is a subset of $D(all)$. We apply a dictionary-based method to measure how informative a term t is in $D(rel)$ against $D(all)$, coined as $w(t)$. Given a query q and a document d, a relevance score can be generated by taking query terms $q = \{t_1^q, t_2^q, ..., t_m^q\}$ and document terms $d = \{t_1^d, t_2^d, ..., t_n^d\}$ into consideration. Specifically, terms are mapped to an L-dimension embedding, namely $vec(.)$. Afterwards, the cosine similarity is employed to compute embedding similarity between two terms, coined as $sim(.,.)$. The ultimate query consists of two parts, $q = \{q_e, q_i\}$, where q_e and q_i stand for explicit and implicit queries, respectively. Based on K-NRM, the relevance scores related to the explicit query $rel(q_e, d)$ and the implicit query $rel(q_i, d)$ are generated, which are aggregated to produce the ultimate score $rel(q, d)$.

B. Explicit Query Formulation

Intuitively, the treatment aspect can be addressed by adding terms that frequently appear in context with explicit expression of information about treatment [3][4]. For instance, Goodwin et al. expand the original query with a list of manually selected terms as listed in Table II [3], in order to emphasize on retrieving biomedical articles containing information about treatment of diseases, as specified in the query.

TABLE II
A snippet of the dictionaries automatically derived from the PM collection using two different term weighting models, as opposed to the manual keyword list of Goodwin et al. [3].

Model	Dictionary			
[3]	prevention	prophylaxis	prognosis	outcome
	survival	treatment	therapy	personalized
Bo1	mutation	egfr	cancer	pten
	braf	protein	kinase	alk
	genetic	tumor	lung	kra
	oncogene	cell	patient	nsclc
KL	mutation	egfr	cancer	pten
	protein	genetic	braf	kinase
	alk	lung	cell	tumor
	patient	oncogene	pancreat	kra

Similar to [3], in our approach, the manually selected treatment-related terms are added to the initial query to formulate the explicit query. Thereby, the explicit query consists of three parts, namely the disease and genetic variants fields in the original query, and the manually added treatment terms. The former two parts are assumed to be more important indicators of the patient' conditions, especially their diseases [6]. Therefore, the three parts of an explicit query are assigned with different weights to reflect their importance to a patient's information need. In particular, akin to [6], disease, genetic variants and terms indicating treatment are assigned with term weights of 3.0, 2.0, and 1.0, respectively. The relevance matching of the explicit query is encapsulated within a neural retrieval framework, as presented in Section II-D.

The manual selection of treatment-related terms, while being able to improve over classical probabilistic retrieval baselines [3], ignores the terms and concepts that are highly likely to appear in the surrounding context of the explicit query terms, such as "tumor" and "kinase", leaving potential for further improvement. Therefore, in the following section, we propose to automatically mine the terms and concepts that are implicitly related to the relevant context.

C. Automatic Implicit Query Generation

In this section, we propose a novel approach to the automatic implicit query generation, by mining the common concepts exist in relevant biomedical articles in response to different user requests. Given the training data composed by the relevant documents judged for a set of queries, we mine a dictionary of implicit concepts that are related to the common information need of PM queries by the following steps.

- **Initial term filtering.** Terms that are too frequent or too rare are supposed to contain too little (e.g. "the") or too specific information, thereby affect the generalization ability of our model over different queries. By taking the within collection frequency as the main feature, the skew model is utilized to rank the terms in the collection such that frequent or rare terms are dropped [7]. Specifically, only

terms whose ranking by the within collection frequency fall within the interval $(s \cdot \#terms, u \cdot \#terms)$ are selected as candidate terms for our implicit dictionary, where $\#terms$ is the vocabulary size of the collection. s and u are set to 0.00007 and 0.001, respectively, as suggested in [7].

- **Term selection.** After ruling out the extremely frequent or rare terms, we select terms to be added to the implicit query by their term weights. Our term weighting technique is based on the Divergence From Randomness (DFR) framework [8], which measures importance of a term by the divergence of its distribution in the relevant document(s) from a random distribution. In this work, the random distribution of a candidate term is approximated by its distribution in the whole document collection. For each term candidate t, we measure its divergence from $D(rel)$ against $D(all)$, namely the divergence of t's distribution in the known relevant documents from its distribution in the whole collection. Put differently, the divergence estimates the degree of t being a common concept out of an assembly of relevant documents of different PM requests. To differentiate between query terms with varied importance, the divergence weight is used as the term weight $w(t)$ in the relevance estimation of the implicit query, as indicated in Equation 5. In particular, two DFR weighting models, KL and Bo1, are used in this work.

In the realm of information retrieval, a commonly used measure for term weighting is the Kullback-Leibler (KL) divergence from a term's distribution in a set of documents against its distribution in the whole collection [9]. Using the KL model, the weight of a term t in $D(rel)$ is given by:

$$w(t) = P_x \cdot \log_2 \frac{P_x}{P_c} \tag{1}$$

where $P_x = tf_x/l_x$ and $P_c = tf_{all}/token_c$. tf_x is the frequency of the term t in $D(rel)$. l_x is the sum of length of each document in $D(rel)$. tf_{all} is the frequency of a term in the whole collection $D(all)$. $token_c$ is the number of tokens in $D(all)$.

Another option for the implicit term weighting is the Bo1 model based on the Bose-Einstein statistics given by the geometric distribution [8]. Under the Bo1 model, the weight of a term t in $D(rel)$ is given by:

$$w(t) = tf_x \cdot \log_2(1 + \frac{1}{tf_{avg}}) + \log_2(1 + tf_{avg}) \tag{2}$$

where tf_{avg} is the mean of the assumed Poisson distribution of term t in the whole collection $D(all)$, which is measured by tf_{all}/N_{all}. N_{all} is the number of documents in the collection. As opposed to KL, Bo1 takes frequency tf_x of individual terms into consideration such that terms receive weights according to their popularity in the relevant document set.

After term weighting, the top k terms are selected to formulate the implicit query, targeting on the common aspects in relevant documents with respect to different PM requests. Both the KL and Bo1 weighting models are used in our experiments.

A snippet of the dictionary automatically derived from the PM collection is shown in Table II. It can be seen that, although the terms generated by our proposed approach are not explicitly related to the treatment information, they are common concepts that are considered important in the relevant context by the term weighting models, and thereby, provide implicit evidence of treatment information, such that the retrieval performance can be improved, as shown in our experiments in Section III.

D. Neural Relevance Model

Relevance matching. Given the generated explicit query and implicit query, the individual relevance score corresponding to each part is obtained by a neural retrieval module without weight sharing. In this paper, the widely-used unsupervised ranking method BM25 [2] serves to produce the initial document ranking; meanwhile the state-of-the-art neural IR relevance matching models, namely, K-NRM [5], is extended for the relevance estimation of both explicit and implicit queries.

As shown in Figure I, the K-NRM model is used to estimate the relevance score of a given document in response to both explicit and implicit queries, respectively. For a quick recap, K-NRM is an interaction-based neural ranking model using Gaussian kernels for the relevance matching of term pairs [5]. Given a query q of length m and a document d of length n, individual terms are mapped into the L-dimension space. Then the similarity of every query-document term pair is computed, results in a similarity matrix M of size $m \times n$. Specifically, each entry (i, j) in matrix M is

$$M_{ij} = sim(vec(t_i^q), vec(t_j^d))$$

Cosine is used for the similarity metric. After that, K-NRM uses f Gaussian kernels to obtain the soft match between each query term and terms in a document, which is given by

$$K_k(M_i) = \sum_j \exp(-\frac{(M - u_k)^2}{2\sigma_k^2}) \tag{3}$$

where u_k and σ_k are the mean and variance of kernel k, respectively. Each kernel estimates how the similarities of term pairs are distributed around the kernel region (u_k, σ_k). The more similarity values close to its mean value, the higher its value. The variance controls the pooling strength of the kernel.

Subsequently, the kernel vector for query term t_i is composed by the f kernels as shown in Equation 4.

$$\vec{K}(M_i) = \{K_1(M_i), K_2(M_i), ..., K_f(M_i)\} \tag{4}$$

The m kernel vectors for the query is then summed up after logarithmic transformation, where query term weights are used as coefficients, as shown in Equation 5.

$$\phi(M) = \sum_{i=1}^{m} \log \vec{K}(M_i) \cdot w(t_i^q) \tag{5}$$

Ultimately, $\phi(M)$ is fed into a learning to rank layer to produce the relevance score as follows.

$$f(q, d) = \tanh(w^T \phi(M) + b) \tag{6}$$

Aggregation of the relevance signals. K-NRM is adopted for the relevance weighting in response to both explicit and implicit queries, before being aggregated to produce the final relevance score. Specifically, a dense layer is employed to aggregate the relevance scores from explicit query $rel(q_e, d)$ and implicit query $rel(q_i, d)$. Intuitively, $rel(q_e)$ directly measures explicit matching signals. Meanwhile, as a complement, $rel(q_i)$ estimates to what extent the candidate document reflects the treatment aspect by matching to concepts that are in common for different PM requests. As a result, the model is learned to weight these two signals and generate an ultimate score $rel(q, d)$.

E. Loss Function and Model Training

Each training sample consists of a query q, a relevant target document d^+ and a non-relevant target document d^- from the ground truth. A hinge loss is employed for training as follows.

$$\mathcal{L}(q, d^+, d^-) = max(0, 1 - rel(q, d^+) + rel(q, d^-))$$

Adam [10] is employed for optimization where batch size is set to 32. The learning rate η is set to 0.001. In addition, early stopping [11] is applied to prevent over-fitting.

III. Experiment

In this section, we conduct experiments to compare the proposed K-NRM^{PM} with multiple state-of-the-art methods from both unsupervised and supervised IR models.

A. Experimental Setting

Dataset. The experiments are based on the standard test collection from biomedical article search task, TREC Precision Medicine (PM) Track 2017. The collection is collected from two different sources: one from a January 2017 snapshot of PubMed abstracts, and the other from the AACR and ASCO proceedings. Apart from PubMed, arguably the largest collection of biomedical documents, the latter is considered to be a likely source of high-quality articles on precision medicine studies. There are a total of 26,739,715 scientific abstracts. The *title*, *abstract* fields are extracted for both sources of documents. For PubMed abstracts, we additionally extract *MeSH Terms*, *journal title*, and *Chemical Compounds* fields. The extracted texts are indexed after Porter stemming and stopword removal [12], resulting in an inverted index with an average document length of 115 tokens. All indexing and unsupervised retrieval experiments are conducted using the open source Terrier toolkit [13]. There are 30 query topics from the TREC 2017 PM Track. Each topic, as shown in the example in Table I, consists of several fields. We use the *disease* and *genetic variants* fields of each topic for retrieval as the other fields are less relevant to the information need.

Competing methods. The following information retrieval models are involved in the experiments.

- **BM25** is a classical unsupervised weighting function based on the two-Poisson distribution of term frequencies in documents [2]. Akin to recent studies on neural IR models [14], BM25 is also used to produce the initial ranking for our proposed K-NRM^{PM}.
- **K-NRM** [5], the state-of-the-art neural retrieval approach, serves another baseline in this study. Due to the lack of training data compared with the commercial data used by [5], we employ a K-NRM variant with a frozen word embedding layer[2], as proposed in [15].
- **Variants of K-NRM$^{\mathbf{PM}}$** are evaluated, including the one using only on the explicit query, coined K-NRM$^{\mathbf{PM}}_{\mathbf{e}}$, and the one using both explicit and implicit queries, coined K-NRM$^{\mathbf{PM}}_{\mathbf{ei}}$. Both Bo1 and KL models are used for the term weighting, as explained in Section II-C. Moreover, K-NRM$^{\mathbf{PM}}(\lambda)$, interpolation of the K-NRM^{PM} variants with BM25, is also evaluated to examine if the extra relevance signals digested in the proposed neural model are beneficial, in addition to the unsupervised baseline.

Metrics. Following the TREC 2017 PM track [1], we employ the following three metrics in the evaluation. The inferred normalized discounted gain (infNDCG) [16] of the top-100 documents is considered to be an accurate evaluation metric when judging a relatively small number of documents while being able to distinguish multiple levels (definitely relevant, possibly relevant, not relevant) of relevance. In addition, R-Precision (R-Prec) and P@10 are also reported for a comprehensive comparison of different methods. The latter two metrics only consider binary relevance. Significant difference is reported based on 95% confidence interval (p-value < 0.05) from a two-tailed Student's t-test.

Cross validation. Due to the limited amount of labeled data, all results are reported based on a 5-fold cross validation. Models are tuned according to infNDCG. All topics in the dataset are split into 5 equal partitions by the order of integer query id. In each fold, we use 3 partitions for training, one for validation, and one for testing. The final results are averaged over all 5 test partitions. Moreover, the implicit dictionary is generated from the training partitions in each fold.

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 [2]. Grid search is applied to tune b (as in [2]) and the number of implicit query terms k. The K-NRM components on the explicit query and implicit query are independent of each other without weight sharing.

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 [17]. The implementation of Word2Vec[3] from [18] is employed. In particular, we employ the skip-gram model, set the dimension to 100, the subsampling threshold to 10^{-3}, and train for 20 iterations.

B. Results

Comparison to BM25. We first compare the retrieval performance of our proposed K-NRM^{PM} models with the unsupervised BM25 and the state-of-the-art neural IR model,

[2]In other words, the word embeddings are not updated during training.
[3]https://code.google.com/p/word2vec/

TABLE III
COMPARISON TO THE UNSUPERVISED BM25 AND THE NEURAL K-NRM BASELINES.

Method	R-Prec		P@10		infNDCG	
$BM25$	0.2615	-	0.4767	-	0.4027	-
K-NRM	0.2694	-	0.4900	-	0.3942	-
K-NRM$^{PM}_{ei(BO)}$	0.2914	8.17%	0.5367	9.53%	0.4275	6.41%
K-NRM$^{PM}_{ei(KL)}$	0.2900	7.65%	0.5300	8.16%	0.4304	7.13%
K-NRM$^{PM}_{ei(BO)}(\lambda)$	**0.3023**	12.21%	**0.5633**†	14.96%	**0.4451**	10.78%
K-NRM$^{PM}_{ei(KL)}(\lambda)$	0.3009	11.69%	**0.5633**†	14.96%	0.4401	9.54%

TABLE IV
COMPARISON TO K-NRM$^{PM}_{e}$ USING ONLY EXPLICIT QUERY *without interpolation* WITH INITIAL RANKING.

Method	R-Prec		P@10		infNDCG	
K-NRM$^{PM}_{e}$	0.2830	-	0.5367	-	0.4113	-
K-NRM$^{PM}_{ei(Bo1)}$	0.2914	2.97%	0.5367	0.00%	0.4275	3.94%
K-NRM$^{PM}_{ei(KL)}$	0.2900	2.47%	0.5300	-1.25%	0.4304	4.64%

TABLE V
COMPARISON TO K-NRM$^{PM}_{e}$ USING ONLY EXPLICIT QUERY *with interpolation* WITH INITIAL RANKING.

Method	R-Prec		P@10		infNDCG	
K-NRM$^{PM}_{e}(\lambda)$	0.2890	-	0.5433	-	0.4443	-
K-NRM$^{PM}_{ei(Bo1)}(\lambda)$	**0.3023**	4.60%	**0.5633**	3.68%	**0.4451**	0.18%
K-NRM$^{PM}_{ei(KL)}(\lambda)$	0.3009	4.12%	**0.5633**	3.68%	0.4401	-0.95%

K-NRM, in Table III. In this table, the best result in each column is in bold. The improvement of our K-NRM^{PM} models over the better baseline is given in percentage, and statistical improvements are marked with †. From Table III, it can be seen that the proposed K-NRM^{PM} variants outperform both baselines by a margin, evaluated by all three different metrics. In particular, when interpolated with the initial relevance score produced by BM25, our approach obtains statistically significant improvement over the best baseline in terms of early precision, measured by P@10. This observation indicates that our proposed neural retrieval framework is able to properly consume relevance signals that are different from the query term matching (as in the baselines) in nature, such that further improvement can be achieved by fusion with the initial ranking. Moreover, the neural retrieval models outperform BM25 in most cases, showing the benefit brought by the adoption of the neural retrieval paradigm.

Comparison to K-NRM^{PM} using only explicit query. In order to examine if the automatically mined implicit dictionary, as presented in Section II-C, is beneficial in addition to the empirically selected manual query, we compare the retrieval performance of our proposed K-NRM^{PM} models with K-NRM^{PM} using only the explicit query, without and with interpolation with the initial results, in Tables IV and V, respectively. From Table IV, it can be seen that the addition of the mined implicit dictionary brings further improvement over K-NRM$^{PM}_{e}$ in terms of R-Prec and infNDCG, which measures the overall performance of top-1k returned documents, but not in terms of early precision measured by P@10. The results indicate that the manually created explicit query can lead to sufficient retrieval accuracy in the top-10 results; meanwhile, the automatically generated implicit query can bring extra performance gain by matching the concepts in common for different PM requests, within a neural retrieval framework. According the results in Table V, obtained by interpolation with the initial ranking, performance improvement of the K-NRM$^{PM}_{ei}$ variants over using only explicit query can be observed in terms of R-Prec and P@10. This again confirms our suggestion that the automatically mined implicit concepts are common in relevant documents for different user queries, such that the fusion with the unsupervised baseline can further enhance the retrieval performance.

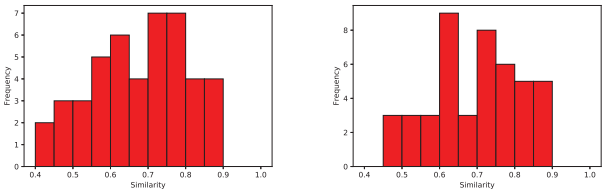
Sanity check. An underlying assumption of our approach is that there exists common terms and concepts that are likely to stand out from relevant documents with respect to different PM queries. To validate this assumption, we conduct sanity check by examining the cosine similarities between implicit dictionaries mined from different random samples. Specifically, we create 10 random samples from the TREC 2017 PM dataset, with each random sample having 1/3 of all the queries, and the overlap between any two random samples is no more than 30%. The distribution of the cosine similarities between the top-50 terms extracted from pairs of random samples is plotted in Figure 2. From Figure 2, it can be seen that the similarities are at least 0.40, mostly beyond 0.60. The average cosine similarities between the implicit dictionaries extracted by Bo1 and KL are 0.696 and 0.672, respectively, which represent fairly strong associations. Indeed, in our additional investigation on traditional information retrieval tasks, such as ad-hoc search on the Robust04 newswire collection [19], such similarities are rarely beyond 0.01.[4] This finding confirms our assumption that the PM task is substantially different from traditional IR tasks. In PM task, there exist relevant medical concepts that are in common among various query topics, but are not explicitly specified in the PM query. Therefore, the retrieval performance of PM systems can be improved by mining those common concepts, and subsequently, incorporating them within a unified neural retrieval framework.

IV. RELATED WORK

The Precision Medicine (PM) Track is a successor of the Clinical Decision Support (CDS) Track [20]. Previous works in CDS have been focusing on query reformulation or query expansion by utilizing the MetaMap knowledge base to map original query to the Unified Medical Language System (UMLS) concepts [21][22][23]. Pseudo relevance feedback has also shown to be effective [23][24].

Goodwin et al. construct a knowledge graph encoding the relationships between drugs, genes, and mutations to incor-

[4]Those results are omitted for brevity.



(a) Bo1 (b) KL

Fig. 2. Distribution of cosine similarities between the top 50 weighted implicit terms mined from different random samples of topics using Bo1 and KL.

porate an aspect-based retrieval paradigm [3]. In particular, they use a manually created treatment keyword list (see Table II) to filter out documents without any of the words on the list. Pasche et al. use positive and negative keyword matching schema to filter the documents and hierarchical query expansion to expand more general (super-type) or more specific (sub-type) cancer type [4]. Pablo López-García et al. create a framework by ElasticSearch to compose queries modularly where final queries were built using blocks called query templates and query decorators [25].

V. Conclusions and Future Work

In this paper, we propose a novel end-to-end K-NRM^{PM} model for the precision medicine (PM) task, in which the relevance scores for both the explicit and implicit queries are generated within a neural retrieval framework. In particular, the implicit query is composed of a list of common treatment concepts mined from documents relevant to different training queries. Experimental results on the standard TREC 2017 PM track dataset indicate superior performance of the proposed K-NRM^{PM} model over the classical unsupervised BM25 model [2] and the state-of-the-art supervised K-NRM model [5]. In addition, comparison to the K-NRM^{PM} model using only explicit query shows extra performance gain brought by the inclusion of the mined implicit query. In the future, we plan to investigate the influence of various neural IR models, such as HiNT [26], Co-PACRR [15], and NPRF [27], for the instantiation of our proposed model.

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

References

[1] K. Roberts, D. Demner-Fushman, E. M. Voorhees, W. R. Hersh, S. Bedrick, A. J. Lazar, and S. Pant, "Overview of the TREC 2017 precision medicine track," in *TREC*, vol. Special Publication 500-324. National Institute of Standards and Technology (NIST), 2017.

[2] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," in *TREC*, vol. Special Publication 500-236. National Institute of Standards and Technology (NIST), 1995.

[3] T. R. Goodwin, M. A. Skinner, and S. M. Harabagiu, "UTD HLTRI at TREC 2017: Precision medicine track," in *TREC*, vol. Special Publication 500-324. National Institute of Standards and Technology (NIST), 2017.

[4] E. Pasche, J. Gobeill, L. Mottin, A. Mottaz, D. Teodoro, P. V. Rijen, and P. Ruch, "Customizing a variant annotation-support tool: an inquiry into probability ranking principles for TREC precision medicine," in *TREC*, vol. Special Publication 500-324. National Institute of Standards and Technology (NIST), 2017.

[5] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *SIGIR*. ACM, 2017, pp. 55–64.

[6] C. Li, B. He, Y. Sun, and J. Xu, "UCAS at TREC-2017 precision medicine track," in *TREC*, vol. Special Publication 500-324. National Institute of Standards and Technology (NIST), 2017.

[7] F. Cacheda, V. Carneiro, V. Plachouras, and I. Ounis, "Performance analysis of distributed information retrieval architectures using an improved network simulation model," *Inf. Process. Manage.*, vol. 43, no. 1, pp. 204–224, 2007.

[8] G. Amati, "Probability models for information retrieval based on divergence from randomness," Ph.D. dissertation, University of Glasgow, UK, 2003.

[9] C. Lioma, B. He, V. Plachouras, and I. Ounis, "The university of glasgow at CLEF 2004: French monolingual information retrieval with terrier," in *CLEF*, ser. Lecture Notes in Computer Science, vol. 3491. Springer, 2004, pp. 253–259.

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

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

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

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

[14] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *CIKM*. ACM, 2016, pp. 55–64.

[15] K. Hui, A. Yates, K. Berberich, and G. de Melo, "Co-pacrr: A context-aware neural IR model for ad-hoc retrieval," in *WSDM*. ACM, 2018, pp. 279–287.

[16] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[17] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *ACL (1)*. The Association for Computer Linguistics, 2016.

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

[19] E. M. Voorhees, "Overview of the TREC 2004 robust track," in *TREC*, vol. Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.

[20] K. Roberts, M. S. Simpson, E. M. Voorhees, and W. R. Hersh, "Overview of the TREC 2015 clinical decision support track," in *TREC*, vol. Special Publication 500-319. National Institute of Standards and Technology (NIST), 2015.

[21] J. R. M. Palotti and A. Hanbury, "TUW @ TREC clinical decision support track 2015," in *TREC*, vol. Special Publication 500-319. National Institute of Standards and Technology (NIST), 2015.

[22] 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*, vol. Special Publication 500-319. National Institute of Standards and Technology (NIST), 2015.

[23] H. Gurulingappa, L. Toldo, C. Schepers, A. Bauer, and G. Megaro, "Semi-supervised information retrieval system for clinical decision support," in *TREC*, vol. Special Publication 500-321. National Institute of Standards and Technology (NIST), 2016.

[24] Y. Ran, B. He, K. Hui, J. Xu, and L. Sun, "A document-based neural relevance model for effective clinical decision support," in *BIBM*. IEEE Computer Society, 2017, pp. 798–804.

[25] P. López-García, M. Oleynik, Z. Kasác, and S. Schulz, "TREC 2017 precision medicine - medical university of graz," in *TREC*, vol. Special Publication 500-324. National Institute of Standards and Technology (NIST), 2017.

[26] Y. Fan, J. Guo, Y. Lan, J. Xu, C. Zhai, and X. Cheng, "Modeling diverse relevance patterns in ad-hoc retrieval," in *SIGIR*. ACM, 2018, pp. 375–384.

[27] C. Li, Y. Sun, B. He, L. Wang, K. Hui, A. Yates, L. Sun, and J. Xu, "NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval," in *EMNLP*. Association for Computational Linguistics, 2018.