Framing Electronic Medical Records as Polylingual Documents in Query Expansion

Edward W Huang\textsuperscript{1}, Sheng Wang\textsuperscript{1}, Doris Jung-Lin Lee\textsuperscript{1}, Runshun Zhang\textsuperscript{2}, Baoyan Liu\textsuperscript{3}, Xuezhong Zhou\textsuperscript{4}, ChengXiang Zhai\textsuperscript{1}

\textsuperscript{1}University of Illinois at Urbana-Champaign, Urbana, Illinois, USA; \textsuperscript{2}Guang’anmen Hospital, Beijing, China; \textsuperscript{3}National Data Center of Traditional Chinese Medicine, Beijing, China; \textsuperscript{4}Beijing Jiaotong University, Beijing, China

Abstract

We present a study of electronic medical record (EMR) retrieval that emulates situations in which a doctor treats a new patient. Given a query consisting of a new patient’s symptoms, the retrieval system returns the set of most relevant records of previously treated patients. However, due to semantic, functional, and treatment synonyms in medical terminology, queries are often incomplete and thus require enhancement. In this paper, we present a topic model that frames symptoms and treatments as separate languages. Our experimental results show that this method improves retrieval performance over several baselines with statistical significance. These baselines include methods used in prior studies as well as state-of-the-art embedding techniques. Finally, we show that our proposed topic model discovers all three types of synonyms to improve medical record retrieval.

Introduction

The Obama administration made electronic medical records (EMR) a high-priority initiative, devoting significant amounts of resources to improve their adoption rate in healthcare practices\textsuperscript{1}. As EMR databases are further introduced into daily usage, retrieval systems are increasing in importance. In particular, one important application of an EMR retrieval system is to efficiently parse medical records and identify those that are most relevant to a new patient. Standard information retrieval systems receive string queries as input, compute numeric scores that determine how well each database document matches the query, and output a ranked list of documents. Ideally, a doctor can query a system with a new patient’s symptoms and receive a set of relevant patient records. These records can serve as an informative baseline to prescribe suitable treatments for the new patient.

However, due to synonyms that occur in the medical vocabulary, the original search query may not be complete, and thus may not retrieve optimal results. There are three categories of synonyms of interest to medical record retrieval:

1. **Semantic synonyms** are medical terms that have identical meanings. For instance, “halitosis” and “fetor oris” are semantic synonyms because they are different terms that refer to the same symptom. Because doctors will typically record only one of these, queries that contain “halitosis” will not properly match records that contain “fetor oris”, and vice versa. Semantic synonyms can be mined with natural language processing techniques and straightforward statistical measures\textsuperscript{2}.

2. **Functional synonyms** are terms that are not identical, but co-occur more frequently than random. “Arthritis” and “hypertension” are functional synonyms due to their comorbid relationship (a decade-long study showed that nearly half of elderly arthritis patients also suffered from hypertension)\textsuperscript{3}. Thus, if a hypothetical query consists of only the term “arthritis” to describe an elderly arthritis patient with hypertension, the retrieved patient records may not contain treatments optimally suited for the query patient. Functional synonyms can be inferred from treatment synonyms, which are described next.

3. **Treatment synonyms** are drug-symptom pairs in which the drug treats the symptom. For example, “ibuprofen” and “fever” are treatment synonyms. Treatment synonyms can be obtained by mining medical publications\textsuperscript{4}.

If a query consists entirely of symptoms, then semantic and functional synonyms are also symptoms. We show that augmenting an original patient query with all three synonym types can capture relevant but mismatched documents, thus improving retrieval performance.
In prior work, Zeng et al. performed query expansion in a similar medical record retrieval setting using synonyms and topic models. Their synonym-based query expansion utilized the Unified Medical Language System (UMLS), which is a compendium of biomedical vocabularies, to map query terms to their semantic synonyms. We refer to this method as the dictionary-based query expansion to avoid confusion. On the other hand, their topic-model-based query expansion trained on patient records to jointly find all three synonym types. However, using this standard topic model, symptoms and treatments are grouped together with no distinction in the medical records, which may decrease performance.

Rather than jointly mine these three types of synonyms, we separately modeled the symptom and treatment synonyms. Our approach is based on traditional monolingual topic models, but instead views the symptoms and treatments of a medical record as generated by distinct languages. Thus, outputted topics will be aligned across the two languages and will contain synonyms of all three types. This is because symptoms in the same topic are likely to be semantic or functional synonyms, while symptoms and treatments in aligned topics are likely to be treatment synonyms. Synonyms of query symptoms can then be used to augment the original query during retrieval. To the best of our knowledge, our proposed method is the first to model symptoms and treatments as separate languages in electronic medical records. We also compare with two embedding methods that jointly mine all three synonyms.

We evaluated our approach on a traditional Chinese medicine (TCM) medical record collection. We chose this dataset because functionally synonymous symptoms are even more prevalent in the TCM field. Thus, if our method could improve retrieval performance on this dataset, it would also work well for EMR datasets in other domains. We show that our method can improve over baseline methods, as well as state-of-the-art embedding methods, in query expansion.

**Problem Formulation**

Given a database of patient records \( R = \{ r_1, \ldots, r_n \} \), the \( i \)th patient record \( r_i \) consists of a set of diseases \( D_i \), a set of symptoms \( S_i \), and a set of treatments \( T_i \). From this database, we wish to retrieve the set of patient records most relevant to some new patient \( p_{new} \) who is not in the database. To achieve this, we first reformulate \( p_{new} \)'s symptoms as a query. Thus, given \( p_{new} \)'s set of symptoms \( S_{new} = \{ s_1, \ldots, s_j \} \),

\[
Q_{new} = S_{new} = \{ s_1, \ldots, s_j \}
\]  

(1)

By performing query expansion on \( Q_{new} \), we add query terms to better match relevant records and thus improve the retrieval performance:

\[
Q'_{new} = \{ s_1, \ldots, s_j, q_1, \ldots, q_l \}
\]  

(2)

Here, \( \{ q_1, \ldots, q_l \} \) is the set of expansion terms that are added to the original query. Although the original query \( Q_{new} \) only contains symptoms, the expansion terms can contain both symptoms and treatments. Expansion terms can be obtained with a variety of methods, which we discuss in the next section.

We hypothesize that the expanded query, \( Q'_{new} \), will retrieve more relevant documents because in practice, \( Q_{new} \) is usually not comprehensive. In our medical setting, this is analogous to situations in which the list of symptoms that a doctor identifies in a new patient is incomplete, which may be due to a combination of two major factors.

1. The doctor uses one of many possible synonyms, including semantic, functional, and treatment synonyms, to describe a patient’s condition.
2. The database is incomplete, so a query symptom may simply not appear in existing medical records, resulting in poor query matches.

We expect the first factor to have a larger impact on query quality, particularly due to unique variations of symptoms that are prevalent in TCM.

**Methods**

With each technique that we used in our experiments, we conducted query expansion, which is a form of pseudo-relevance feedback, to improve retrieval performance. We augmented each query with synonym terms selected by
each method, and then performed the retrieval on the existing database of medical records.

Overall, we used five different methods of query expansion. First, we addressed two baselines used in previous work\(^5\): the dictionary-based query expansion and the topic-model-based query expansion. In our dataset, the dictionary-based method incorporated an external treatment-symptom knowledge graph to add manually curated treatment synonyms. The topic-model-based method trains topics on the patient record database to add expansion terms that co-occur in the same topics as the given query. Although the previous study used a third method, predicate-based query expansion, we did not utilize this method due to a lack of high-quality TCM ontology databases. Furthermore, the predicate-based method was unable to outperform the topic-model-based method in prior work.

Next, we explored two network embedding techniques, Med2Vec and diffusion component analysis (DCA). Med2Vec is an embedding algorithm that learns efficient representations of medical records and concepts by using EMR datasets. On the other hand, DCA performs network embedding on the knowledge graph utilized in dictionary-based query expansion to obtain vector representations of nodes in the graph. The key difference between these two methods is that Med2Vec does not depend on expert medical knowledge, while DCA does.

Lastly, we discuss our method, which mines semantic, functional, and treatment synonyms by considering symptoms and treatments to be separate languages in a topic model.

**Dictionary-Based Query Expansion**

Dictionary-based query expansion utilizes a ground-truth, treatment-symptom TCM dictionary. This dictionary was curated from a TCM textbook containing known relations, interactions, and treatments. For example, the “crow-dipper” herb has multiple entries, treating symptoms from vertigo to breathing difficulties. We constructed a knowledge graph, in which an undirected edge \( \{t, s\} \) indicates that a treatment \( t \) treats a symptom \( s \) in the dictionary. There are 1,995 treatments, 1,635 symptoms, and 27,824 treatment relations in the dictionary, which translated to a total of 3,630 nodes and 27,824 edges in the resulting knowledge graph. There are no treatment-treatment or symptom-symptom edges. To perform query expansion on a query \( Q_{\text{new}} \), we add all treatments from the knowledge graph that are directly connected to at least one symptom in \( Q_{\text{new}} \).

**Topic-Model-Based Query Expansion**

In prior work, topic-model-based query expansion performed the best in a similar medical record retrieval task in terms of recall and F-measure\(^5\). Specifically, the authors used latent Dirichlet allocation (LDA)\(^7\) to derive topics from their database of EMRs. With LDA, each document is characterized by a mixture of topics. In turn, each topic consists of mixtures of words. In our study, we also used LDA to train topics from the dataset.

After training \( k \) topics, from topic \( i \)’s per-word distribution \( \phi_i \), we refer to the set of 100 words with the highest probabilities as \( H_i \). For a query \( Q_{\text{new}} \), we then perform the following multiplication:

\[
\phi'_i = |Q_{\text{new}} \cap H_i| \cdot \phi_i
\]  

(3)

With this operation, we scale each word’s probability in \( \phi_i \) by the number of query terms that are in the top 100 words of \( \phi_i \). Finally, we sum each word’s probabilities across the scaled distributions, \( \sum_{i=1}^{k} \phi'_i \), and receive a new weight for each word. We experimentally chose to identify five topics from our dataset. The top five words with the highest weights were designated expansion terms.

**Med2Vec-Based Query Expansion**

Med2Vec is a state-of-the-art embedding method designed specifically for EMRs\(^8\). It discovers efficient representations of “medical codes” (symptoms and treatments, in the case of our dataset). To learn embedding from patient records, Med2Vec’s optimization function is similar to that of word embedding methods that use the skip-gram model, such as word2vec\(^9\). The authors stated three major reasons for word2vec’s failure to accommodate medical data:

1. Healthcare datasets have unique structures in which the visits are temporally ordered, but the medical codes
within a visit form an unordered set.

2. Learned representations should be interpretable.

3. The algorithm should be scalable to handle real-world datasets of millions of patients.

In particular, the first reason is of greatest relevance to our experiment setting. Med2Vec maximizes the likelihood of observing a medical code (symptom or treatment) given the codes in the same visit. In other words, a medical code’s vector representation predicts its neighboring medical codes. By obtaining vector representations of all medical codes as well as computing their pairwise similarities, Med2Vec jointly discovers semantic, functional, and treatment relationships.

We ran Med2Vec on our training corpus and obtained a set of low-dimensional vector representations for each symptom and treatment in the dataset. Given a query \( Q_{new} = \{s_1, \ldots, s_j\} \), we computed the cosine similarity between each query term \( s_i \)’s Med2Vec representation and every non-query term’s Med2Vec representation. Thus, for each candidate expansion term, we summed \( j \) similarity scores, one for each query term. We took the five terms with the highest score sums as expansion terms.

**DCA-Based Query Expansion**

DCA is a network embedding technique that takes a network as input and then learns low-dimensional vector representations of the input’s nodes\(^\text{10}\). DCA has been shown to achieve excellent results in learning network structure for gene function prediction\(^\text{11}\).

DCA ensures that two nodes have very similar low-dimensional representations if they are topologically close in the network. Thus, related medical concepts tend to have similar output vectors. Like Med2Vec, DCA jointly mines all three synonym types. We used the network constructed in dictionary-based query expansion as the input to DCA. After learning vector representations for each node in the network, we computed cosine similarity scores as in Med2Vec-based query expansion, again taking the top five terms with the highest score sums as expansion terms.

**BiLDA-Based Query Expansion**

In our data, symptoms and treatments are labeled and separated in each patient record. We hypothesize that a topic model that explicitly captures this structure will improve performance over standard, monolingual topic models.

Polylingual topic modeling (PLTM) finds latent cross-lingual topics in a multilingual corpus\(^\text{12}\). These text collections can either be direct translations or theme aligned\(^\text{13}\). Direct translations occur in sentences of two documents that are meant to be translations of one another. An example of a direct translation is “Romeo and Juliet” in English and Chinese. On the other hand, theme alignment occurs in documents that are not necessarily direct translations, but discuss the same topics in similar sections. An example of theme alignment is the Wikipedia pages on “Romeo and Juliet” in English and Chinese.

Our method considers EMRs to consist of two separate “languages”: symptoms and treatments. Thus, patient records are theme aligned in the sense that a patient’s symptoms and treatments are generated by the same set of diseases. Furthermore, the symptoms and treatments are varied according to the same syndromes, which are latent factors not explicitly stated in the records. Standard monolingual topic models are unable to represent these separate “languages”, since symptoms and treatments are grouped together. This removes the ability to differentiate, and therefore translate, between the two.

The output of the PLTM is a set of cross-lingual topics, including per-document topic distributions and per-topic word distributions in each of the languages. This model assumes that each topic consists of a discrete distribution of words for each language. Thus, there are two language-specific topics, \( \Phi^S \) and \( \Phi^T \), each of which is drawn from its own symmetric Dirichlet distribution with parameters \( \beta^S \) and \( \beta^T \), respectively.

Next, we discuss the generative process. Each EMR is represented as a mixture over topics, and is generated by first sampling from an asymmetric Dirichlet prior with concentration parameter \( \alpha \) and base measure \( m \):
Figure 1: The plate notation of our proposed model, framing electronic medical records as bilingual documents. A variable’s superscript $S$ indicates symptoms and $T$ indicates treatments. $\alpha$ and $\beta$ are the parameters of the Dirichlet priors on the per-document topic distributions and the per-topic word distributions, respectively. $\theta_d$ is the topic distribution for document $d$. $\phi^S_k$ and $\phi^T_k$ are each language’s corresponding word distributions for topic $k$. $z$ is the latent topic assignment for each observed word $w$.

$$\theta \sim \text{Dir}(\theta, \alpha m)$$ (4)

Then, a latent topic assignment is drawn for each word in the corresponding “language” (i.e., symptoms and treatments).

$$z^S \sim P(z^S | \theta) = \prod_r \theta^S_{z^S_r}$$ (5)

$$z^T \sim P(z^T | \theta) = \prod_r \theta^T_{z^T_r}$$ (6)

The individual symptoms and treatments are then drawn using language-specific topic parameters.

$$w^S \sim P(w^S | z^S, \Phi^S) = \prod_r \phi^S_{w^S_r | z^S_r}$$ (7)

$$w^T \sim P(w^T | z^T, \Phi^T) = \prod_r \phi^T_{w^T_r | z^T_r}$$ (8)

With two languages, PLTM reduces to Bilingual Latent Dirichlet Allocation (BiLDA) (Figure 1). We obtained $k$ joint topics that align $k$ symptom topics and $k$ treatment topics. As with monolingual LDA, we experimented with different values of $k$, finding $k = 5$ to yield the best results. To train topics with BiLDA, we used the MAchine Learning for Language Toolkit (MALLET)\textsuperscript{14}, which performs inference with Gibbs sampling. We conducted query expansion the same way we performed LDA-based query expansion, selecting the five terms with the highest sums.

Evaluation

We evaluated and compared the five different query expansion methods, as well as the baseline with no query expansion, by performing retrieval on our dataset. We first describe our dataset, then discuss the evaluation process in the following two sections.

Data Description

We used a large, real-world EMR database containing 7,553 anonymous patient visits, obtained from the department of gastroenterology at Guang’anmen Hospital in Beijing, China. The same doctor treated all patients in the record,
which has the advantage of consistent treatment, but the simultaneous disadvantage of potentially systematic errors or incompleteness. All patients had some variety of stomach illness. Each record contains a detailed list of symptoms, treatments, and diseases. Each patient had an average of 9.08 symptoms and 1.63 diseases. The disease information was used as ground truth labels in the evaluation stage, and was therefore not included when finding query expansion terms. We elected to use only the first visit for each patient to prevent cases in which a patient’s query returns other visits of the same patient. This left us with 3,750 patient visits.

Cross-Validation
We split our dataset into ten random training and test sets as per $k$-fold cross-validation. Thus, the training records were functionally a database of EMRs. Each held-out test patient was then regarded as a new, unseen patient. For each of the test patients, we retrieved relevant patient documents from the training set.

Each test set contained 375 patient records. We excluded all details from the test set except for symptoms. Using a given test patient’s symptom set as a query, we performed each query expansion method in three ways: adding symptoms, adding treatments, and adding both. We refer to these methods as “symptom expansion”, “treatment expansion”, and “mixed expansion”, respectively. The only exception to this was the dictionary-based query expansion, which is only capable of treatment expansion.

Retrieval Tests
For each query in the held-out test set, we performed medical record retrieval. To score a document in the training corpus given a query patient, we used Okapi BM25 as our ranking function, which is one of the most effective retrieval methods\cite{15}. The Okapi BM25 score of a document $D$ given a query patient $Q = \{q_1, \ldots, q_n\}$ is defined as

$$\text{Score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

(9)

In our experiments, $f(q_i, D)$ was always 1 if $q_i$ appeared in $D$, since no patient record contained duplicate symptoms or treatments. $|D|$ is the length of document $D$, and avgdl is the average document length in the training corpus. For the symptom expansions and the baseline with no query expansion, $D$ contained only symptoms. For treatment and mixed expansions, $D$ contained all symptoms and treatments of the patient. In the absence of parameter optimization, we chose the default values of $k_1 = 2$ and $b = 0.75$. Additionally, the inverse document frequency is defined as

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

(10)

where $N$ is the total number of documents in the training corpus, and $n(q_i)$ is the number of training documents containing the term $q_i$. With this ranking function, we returned a ranked list of retrieved documents given a query $Q$.

Relevance Measure
To evaluate the performance of each retrieval task, we assigned an objective measure of relevance to a retrieved patient given a query patient. Conveniently, the list of diseases the doctor assigned to each patient was recorded in our dataset. We used these disease lists as ground truth labels for the corresponding patients. Thus, we define the gain of a document $D$ given a query patient $Q$ to be the following:

$$\text{Gain}(D, Q) = \frac{|D_{\text{disease}} \cap Q_{\text{disease}}|}{|D_{\text{disease}}||Q_{\text{disease}}|}$$

(11)

Here, $D_{\text{disease}}$ and $Q_{\text{disease}}$ refer to the set of diseases belonging to $D$ and $Q$, respectively. In the traditional vector space model, this gain is the cosine similarity between the document and query vectors, which is a useful metric for determining similarity between two documents\cite{16}. Thus, we can use normalized discounted cumulative gain (NDCG),
Table 1: Retrieval results for various query expansion methods. Bolded values indicate the highest NDCG@k. BiLDA mixed expansion performed best for all choices of k.

<table>
<thead>
<tr>
<th>Expansion Method</th>
<th>NDCG@5</th>
<th>NDCG@10</th>
<th>NDCG@15</th>
<th>NDCG@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>No query expansion</td>
<td>0.1673</td>
<td>0.1675</td>
<td>0.1674</td>
<td>0.1677</td>
</tr>
<tr>
<td>Dictionary</td>
<td>0.1633</td>
<td>0.1647</td>
<td>0.1652</td>
<td>0.1659</td>
</tr>
<tr>
<td>LDA (symptoms)</td>
<td>0.1686</td>
<td>0.1682</td>
<td>0.1681</td>
<td>0.1690</td>
</tr>
<tr>
<td>LDA (treatments)</td>
<td>0.1689</td>
<td>0.1669</td>
<td>0.1667</td>
<td>0.1677</td>
</tr>
<tr>
<td>LDA (mixed)</td>
<td>0.1690</td>
<td>0.1671</td>
<td>0.1668</td>
<td>0.1679</td>
</tr>
<tr>
<td>Med2Vec (symptoms)</td>
<td>0.1636</td>
<td>0.1637</td>
<td>0.1648</td>
<td>0.1652</td>
</tr>
<tr>
<td>Med2Vec (treatments)</td>
<td>0.1682</td>
<td>0.1673</td>
<td>0.1684</td>
<td>0.1677</td>
</tr>
<tr>
<td>Med2Vec (mixed)</td>
<td>0.1678</td>
<td>0.1671</td>
<td>0.1685</td>
<td>0.1675</td>
</tr>
<tr>
<td>DCA (symptoms)</td>
<td>0.1518</td>
<td>0.1534</td>
<td>0.1556</td>
<td>0.1560</td>
</tr>
<tr>
<td>DCA (treatments)</td>
<td>0.1689</td>
<td>0.1702</td>
<td>0.1712</td>
<td>0.1719</td>
</tr>
<tr>
<td>DCA (mixed)</td>
<td>0.1510</td>
<td>0.1537</td>
<td>0.1557</td>
<td>0.1565</td>
</tr>
<tr>
<td>BiLDA (symptoms)</td>
<td>0.1709</td>
<td>0.1706</td>
<td>0.1713</td>
<td>0.1716</td>
</tr>
<tr>
<td>BiLDA (treatments)</td>
<td>0.1684</td>
<td>0.1681</td>
<td>0.1681</td>
<td>0.1679</td>
</tr>
<tr>
<td>BiLDA (mixed)</td>
<td><strong>0.1752</strong></td>
<td><strong>0.1739</strong></td>
<td><strong>0.1747</strong></td>
<td><strong>0.1736</strong></td>
</tr>
</tbody>
</table>

a standard method of evaluating search engines\textsuperscript{17}, to compute the quality of our ranked list. The DCG at a particular rank $k$, for query $Q$, which returns a ranked list of $D_1, \ldots, D_N$ is defined as

$$DCG@k = \sum_{i=1}^{k} \frac{Gain(D_i, Q)}{\log_2(i + 1)}$$  \hspace{1cm} (12)

where $Gain(D_i, Q)$ is defined in Equation 11. NDCG@$k$ is defined as DCG@$k$ divided by the DCG of the ideal ranked list for query $Q$, thus making it a metric comparable across queries and suitable for our 10-fold framework. We show results for $k \in \{5, 10, 15, 20\}$. We exclude precision, recall, and the F-measure due to their inability to incorporate rankings.

Results and Discussion

The results of the evaluation are shown in Table 1. In order to analyze the significance of the NDCG values, we performed the paired $t$-test on the ranking metrics between pairs of expansion methods.

BiLDA-based mixed query expansion achieved the best retrieval performance among all expansions. For NDCG@5, 10, 15, and 20, it performed better than the baseline with no query expansion with $p$-values of $2.842 \times 10^{-3}$, $4.784 \times 10^{-3}$, $6.852 \times 10^{-7}$, and $1.929 \times 10^{-6}$, respectively. Furthermore, BiLDA mixed expansion performed better than all of the runner-up methods at the 5\% significance level.

Mixed expansion was only the best-performing expansion type for BiLDA. This is due to the fact that all other methods do not separately mine symptom and treatment synonyms. On the other hand, BiLDA-based query expansion considers symptoms and treatments to be from separate topics, and therefore it successfully added in mixed query terms.

Dictionary-based expansion’s poor performance can be explained by the fact that it adds too many treatment synonyms, which dilutes the original query’s symptoms. On average, dictionary-based expansion nearly doubled each query in size.

We show an example of a mixed query expansion from BiLDA. In the patient query in Table 2, the five expansion terms include three symptoms and two treatment herbs. Fluttering pulse was an expansion term for this patient, and is indeed an indicative symptom of the patient’s disease, chronic gastritis\textsuperscript{18}. Dark, red tongue is a functional synonym of yellow, greasy tongue coating, both of which commonly appear in patients with chronic gastritis\textsuperscript{19}. Fullness is a
Table 2: Example of an expanded query created by the BiLDA method. Different query terms are separated by semicolons.

<table>
<thead>
<tr>
<th>Disease Label</th>
<th>Original Query</th>
<th>Expansion Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chronic gastritis</td>
<td>yellow, greasy tongue coating; epigastric chills; heartburn; bloating; belching; stomachache; acid reflux; dry mouth</td>
<td>fluttering pulse; dark, red tongue; fullness; bitter orange; crow-dipper</td>
</tr>
</tbody>
</table>

...semantic synonym of bloating. Bitter oranges are known to treat abdominal bloating\textsuperscript{20}, and are furthermore known to treat chronic gastritis\textsuperscript{21}. Lastly, crow-dippers are also known to treat bloating in chronic gastritis patients\textsuperscript{22}. Indeed, crow-dipper was actually prescribed to this particular patient.

Related Work

Zeng et al. performed a study of synonym, topic model, and predicate-based query expansions. They used monolingual LDA as their choice of topic model and determined it to be the best-performing method\textsuperscript{5}. Our work builds upon their study in the context of traditional Chinese medicine, while also comparing additional methods, showing BiLDA to be an even more effective method. A major difference between their work and ours is that while both systems aim to return the most similar patients to a query, their experimental queries consisted of a single primary disease (“PTSD” and “diabetes”), while our query consists of the complete set of symptoms per patient. Furthermore, we refine the choice of evaluation from traditional measures of precision, recall, and $F_1$ to the more comparable metric of NDCG@$k$. Choi et al. developed a method of learning efficient representations of medical codes, Med2Vec, which we used as one of this study’s expansion methods\textsuperscript{8}. In a previous work, we also performed DCA on a prior knowledge dictionary, but in the context of patient record matrix enrichment for subcategorization of TCM syndromes\textsuperscript{6}. Jain et al. also performed medical record retrieval with query expansion on a patient’s symptoms\textsuperscript{1}. However, they used a knowledge base by integrating domain ontologies and automatic semantic relationship learning, similar to Zeng et al.‘s predicate-based query expansion. Due to the lack of TCM ontologies, this method was infeasible.

Conclusions and Future Work

In this paper, we studied how medical record retrieval can improve with query expansion. Prior work showed topic-model-based query expansion to perform the best\textsuperscript{5}. We presented an improved topic model that frames symptoms and treatments as distinct languages.

We performed query expansion on EMR retrieval experiments with latent Dirichlet allocation (LDA), a treatment-symptom dictionary, Med2Vec, diffusion component analysis (DCA), and a polylingual topic model. LDA and dictionary synonyms were studied in prior work and thus used as baselines in this paper. Med2Vec is an EMR-specific embedding approach that learns interpretable representations of medical concepts. DCA is a network embedding method that incorporates prior TCM knowledge to also learn low-dimensional representations of medical concepts. Our experimental results showed that our method performs the best by normalized discounted cumulative gain, with significant $p$-values computed by paired $t$-tests.

Future work includes experimenting with other methods of query expansion. For instance, pointwise mutual information (PMI) has shown promising treatment-symptom pairings. Another potential method is the Weighted Exclusivity Test (WExT), which computes triples of medical concepts as an extension to PMI\textsuperscript{23}.

Lastly, a fundamental change to our problem would be to reframe the retrieval task as a treatment recommendation system. Like before, the system would take a test patient’s set of symptoms as the input query. However, instead of retrieving patient records relevant to the query, the system would recommend a set of drugs to treat the query symptoms. We can evaluate the new system by counting the number of recommended treatments that match the actual prescribed treatments for the test patient. With this framework, we can skip the step in which the doctor analyzes the set of returned patients in the retrieval task and instead directly recommend treatment. In fact, the embedding and knowledge graph-based methods, in addition to PMI and WExT outputs, already have explicit treatment-symptom relationships that would enable this new framework.
Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1144245. This research was also supported by grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Our work also received support from National Key Technology R&D Program grants 2013BAI02B01 and 2013BAH06F03, National S&T Major Project of China grant 2013BAI13B04, and Special Research Project of TCMS by State Administration of Traditional Chinese Medicine grant 201407001.

References


