Combining Multi-level Evidence for Medical Record Retrieval

Dongqing Zhu and Ben Carterette
Department of Computer & Information Sciences
University of Delaware
Newark, DE, USA 19716
[zhu | carteret]@cis.udel.edu

ABSTRACT

The increasing prevalence of electronic health records containing rich information about a patient's health and physical condition has the potential to transform research in health and medicine. In this work, we present a health record search system for finding patients matching certain inclusion criteria (specified as keyword queries) for clinical studies. In particular, our system aggregates multi-level evidence and combines proven statistical IR models, both in an innovative way, and achieves a 20% MAP (mean average precision) improvement over a strong baseline. Moreover, our cross-validation results show that the overall performance of our system is comparable to other top-performing systems on the same task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Design, Experimentation

Keywords

medical record search, EHR, information retrieval, evidence aggregation, language models

1. INTRODUCTION

The increasing prevalence of electronic health records (EHR), along with the needs for enhanced clinical care, presents new challenges to information retrieval (IR). Much clinical research work relies on the ability to find relevant health records and gather sufficient clinical evidence under severe time constraints. For example, the identification of a sample of patients for the purpose of a prospective clinical trial, or retroactive evaluation of the effects of particular treatments in the context of a data warehouse. They both necessarily involve a search for patients to include.

SHB'12, October 29, 2012, Maui, Hawaii, USA.

While some searches for similar tasks can be straightforward (practically solvable with SQL-like queries on relational data), many run up against the standard problems of information retrieval: heterogeneity of language, polysemy, synonymy, and all the other problems that computational processing of natural language presents. For example, a query for "patients with hearing loss" will match many nonrelevant documents with references to weight loss, simply because the word "loss" occurs much more often with "weight" than with "hearing".

In this work, we present a system for searching flat-text health records (i.e. the doctors' notes sections of EHR) for patients with particular conditions specified via a keyword query. In particular, our system improves a strong baseline in two aspects. First, it aggregates multi-level evidence in a novel way and significantly boosts its retrieval performance. Second, it innovatively combines two proven statistical IR models to tackle two specific problems of search in medical domain, and further improves the retrieval performance significantly. As our cross-validation results will show, our system achieves a 20% MAP (mean average precision) improvement over a strong baseline. Moreover, the overall performance of our system is comparable to other top-performing systems on the same task.

This paper is organized as follows: Section 2 introduces the retrieval task and data for our experiments. Then, Section 3 describes a basic retrieval model based on language modeling, and shows how we aggregate the multi-level evidence in the medical reports for retrieval. Section 4 details the evaluation metrics and experimental setup. Next, Section 5 presents our experimental results and research findings. Section 6 describes further improvement. Then, Section 7 highlights related work. Finally, Section 8 concludes the paper and points out possible directions for future work.

2. RETRIEVAL TASK AND DATA

We use the official test collection from the TREC 2011 Medical Records Track [16] for our experiments. The test collection contains 100,866 de-identified medical reports from the University of Pittsburgh NLP Repository. These medical reports were gathered from multiple hospitals in the course of one month. The retrieval task[1] is an ad hoc search task for patient visits. A patient visit to the hospital usually results in multiple medical reports, meaning there is a 1-to-n relationship between visits and reports. Based on the report-to-visit mapping information provided with the

[1] `http://www-nlpir.nist.gov/projects/trecmed/2011/tm2011.html`

TREC test collection, we have 17,198 unique visits associated with 100,866 reports.

Each medical report is an XML file with a fixed set of fields[2]. The most important information resides in two diagnosis fields consisting of ICD-9 (International Classification of Diseases, 9th Revision) codes, and one free-text field containing doctors' notes. Our search system will rely on evidence within these fields to satisfy search users' information needs.

TREC assessors developed 35 information needs (or "topics" in TREC terminology). These needs were designed to require information mainly from the free-text fields, i.e., topics are not answerable solely by the diagnostic codes. Topics are meant to reflect the types of queries that might be used to identify cohorts for comparative effectiveness research [16]. Table 1 lists several TREC topics as examples. The topic usually specifies the patient's condition, disease, treatment, etc. Relevance judgments for the topics were also developed by TREC assessors based on the pooled results from TREC participants.

In summary, the retrieval task is to find patients matching certain inclusion criteria for clinical studies based on a set of medical reports.

ID	Topic
107	Patients with ductal carcinoma in situ (DCIS)
118	Adults who received a coronary stent during an admission
109	Women with osteopenia
112	Female patients with breast cancer with mastectomies during admission

Table 1: Example topics of medical records track

3. RETRIEVAL MODEL

Our retrieval system uses a basic "bag-of-words" probabilistic model: the query likelihood language model. This model scores documents for queries as a function of the probability that query terms would be sampled (independently) from an urn containing all the words in that document. Formally, the scoring function is a sum of the logarithms of smoothed probabilities:

$$\text{score}(D,Q) = \log P(Q|D) = \sum_{i=1}^{n} \log \frac{tf_{q_i,D} + \mu \frac{tf_{q_i,C}}{|C|}}{|D| + \mu}, \quad (1)$$

where q_i is the ith term in query Q, n is the total number of terms in Q, $|D|$ and $|C|$ are the document and collection lengths in words respectively, $tf_{q_i,D}$ and $tf_{q_i,C}$ are the document and collection term frequencies of q_i respectively, and μ is the Dirichlet smoothing parameter. "Document" is a broad term that could indicate different granularities of a visit: the text of all reports associated with the visit, a single report from the visit, or just one field within a single report from the visit. In the next section, we describe how we leverage evidence from each of these to come up with a final document score.

[2]http://www.dbmi.pitt.edu/nlp/report-repository

3.1 Multi-level Evidence

3.1.1 Field Level Evidence

As described above, the main fields in a report are the doctor's notes and the fields that contain diagnosis codes. Here we describe how we leverage ICD-9 codes in the language model, and how we remove some extraneous information from doctor's notes.

Code Expansion: The "admit diagnosis" and "discharge diagnosis" fields contain contain ICD-9 codes which, though mainly used for billing purposes, give a high level summary of report content, and whose associated descriptions can provide potentially useful terms for retrieval purpose. Thus, we expand ICD codes with their corresponding descriptions[3]. For instance, we substitute code "428.1" with "LEFT HEART FAILURE". We refer this feature as ICD in the following sections.

Negation Removal: The "report text" filed contains clinical narratives. One distinct feature of clinical narratives is that negation phrases are frequently used to claim the absence of certain conditions or symptoms [1], such as "cannot tell", "not clear", "without evidence", etc. Negations may cause retrieval false positives. For instance, a simple IR system will consider a document with the sentence "The patient comes in with episodes of orthopnea and has ruled out for an acute coronary syndrome." as relevant to the query "acute coronary syndrome". Thus, we use NegEx[4] [7], an open-source clinical negation detection tool, to remove all negated portions of the sentences from the medical records before indexing. For instance, in the above example we will delete the phrase "ruled out for an acute coronary syndrome" from the original report. We refer this feature as NEG in the following sections.

3.1.2 Report Level Evidence

Evidence in a visit may mainly exist in only a small proportion of all the associated reports. This allows us to rely on the strongest evidence of a visit to estimate its relevance. Thus, we use reports as the initial retrieval units (i.e., building an index for reports and applying the retrieval model to each report), and then transform a report ranking into a visit ranking based on the strongest report-level evidence, which is equivalent to using the following report score merging method for ranking visits:

$$\text{score}_{\text{RbM}}(V,Q) = f_{\text{RbM}}(\{\text{score}(r_1^V, Q), \text{score}(r_2^V, Q), ...\}), \quad (2)$$

where r_j^V is a report associated with visit V based on the report-to-visit mapping, $\text{score}(r_j^V, Q)$ is the language modeling score of the report with respect to query Q, and f_{RbM} is the function for aggregating the scores. We will try MAX, SUM, and ANZ for f_{RbM} in Section 5. We name this evidence aggregation strategy Retrieval-before-Merging (RbM). The merging process involved in RbM corresponds to "merging I" in Figure 1.

3.1.3 Visit Level Evidence

Evidence may also spread across multiple reports, especially when the information need is a complex one. Thus, our second strategy to aggregate evidence is to first merge reports from a single visit field by field into a visit document

[3]https://drchrono.com/public_billing_code_search

[4]http://code.google.com/p/negex/

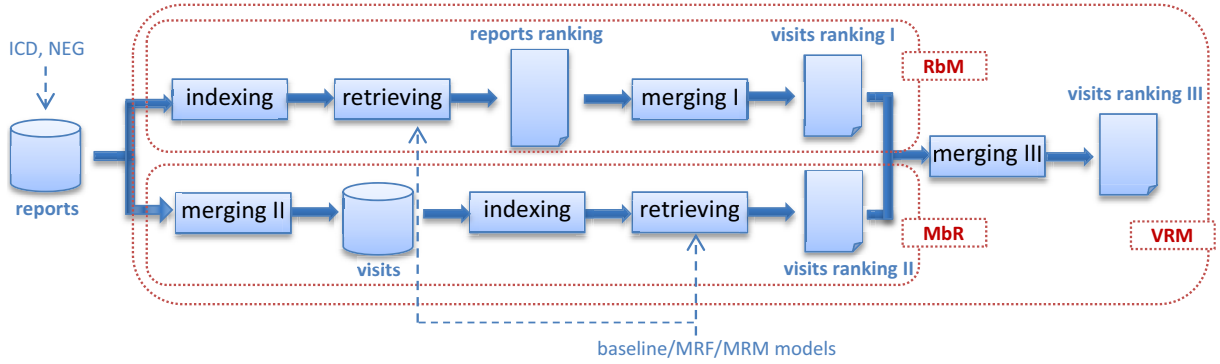

Figure 1: Merging results from two different retrieval methods.

and then construct an index for visit documents. With this strategy, the language model built on a merged document can naturally combine the evidence scattered across multiple reports. Furthermore, this strategy can directly lead to a ranking of visits which are the desired retrieval units. We call this second evidence aggregation strategy Merging-before-Retrieval (MbR). The merging process involved in MbR corresponds to "merging II" in Figure 1.

3.1.4 *Top-level Evidence*

RbM and MbR as described above are two different strategies for aggregating evidence and ranking visits. RbM and MbR complement each other in that the former can naturally aggregate evidence spreading across multiple reports (which would be challenging to do at the report-level) while the latter can leverage the strongest evidence (which may become less apparent after reports merging in MbR) to estimate relevance. This leads to our third evidence aggregation method in which we take advantage of both RbM and MbR by merging their visit rankings, as demonstrated by "merging III" in Figure 1. We call third strategy as Visit-Ranking-Merging (VRM). The merging method (i.e., "Merging III" in Figure 1) is defined by:

$$\text{score}_{\text{VRM}}(V,Q) = f_{\text{VRM}}(\text{score}_{\text{RbM}}(V,Q), \text{score}_{\text{MbR}}(V,Q)), \quad (3)$$

where $\text{score}_{\text{RbM}}(V)$ and $\text{score}_{\text{MbR}}(V)$ are the language modeling scores for visit V with respect to query Q in the two visit rankings obtained by RbM and MbR respectively, f_{VRM} is the function for score aggregation, and $\text{score}_{\text{VRM}}(V,Q)$ is the final score of visit V in the merged ranking. We will try different methods for f_{VRM} such as CombMNZ, CombSUM, and CombMAX in Section 5 below.

4. EVALUATION

This section describes evaluation metrics and experimental setup.

4.1 Evaluation Metrics

The official evaluation metrics for the TREC Medical Records track are precision at rank 10 (P10), bpref, and R-precision (Rprec). Here we also use mean average precision (MAP) as an additional metric. They are defined as follows:

1) P10 measures the proportion of relevant documents among the top 10 retrieved.

2) MAP, as one of the most standard evaluation measures among TREC community, provides a single-figure measure of quality across recall levels [2]. If $\{d_1,...,d_j\}$ is the set of relevant documents for an information need $q \in Q$, then MAP is defined as:

$$\text{MAP}(Q) = \frac{1}{|Q|}\sum_{q\in Q}\frac{\sum_{d\in\{d_1,...,d_j\}}\text{Precision}(\text{rank}(d))}{|\{d_1,...,d_j\}|}, \quad (4)$$

where Precision(k) is the proportion of relevant documents among the top k retrieved.

3) bpref is defined as:

$$\text{bpref} = \frac{1}{R}\sum_{r}(1 - \frac{|n \text{ ranked higher than } r|}{min(R,N)}), \quad (5)$$

where R is the number of judged relevant documents, N is the number of judged irrelevant documents, r is a relevant retrieved document, and n is a member of the first R irrelevant retrieved documents. bpref computes a preference relation of whether judged relevant documents are retrieved ahead of judged irrelevant documents. It is based on the relative ranks of judged documents only.

4) R-precision is the precision after R documents have been retrieved (also known as the break-even point), where R is the number of relevant documents for the topic. It de-emphasizes the exact ranking of the retrieved relevant documents, though it is highly correlated to MAP in practice.

Note that in rest of the paper, when we mention bpref, P10, or Rprec, we are referring to the average score of bpref, P10, or Rprec over all topics in a run.

4.2 Experimental Setup

We use the Indri[5] retrieval system for indexing and retrieving. In particular, we use the Porter stemmer to stem words in both reports and queries, and use a simple standard medical stoplist [8] for stopping words in queries only. Then we conduct 5-fold cross-validation and use top 1000 retrieved visits[6] for each query to evaluate our system under different settings. In each iteration, we train our system on 28 queries to obtain the best parameter setting for MAP by sweeping over the range of [1000, 20000] at a step size of 1000 for the Dirichlet smoothing parameter (i.e., μ in Equation 1), and then generate a ranking for each of the remaining 7 queries based on the trained system. When complete, we have full

[5] `http://www.lemurproject.org/indri/`

[6] The guideline of TREC medical records track requires each retrieval set contain no more than 1000 visits.

rankings for all 35 topics as a test set. We evaluate the system based on the average AP, bpref, P10, and Rprec over all 35 topics.

We train our systems on MAP though bpref is the primary evaluation metric for 2011 medical track. There are two reasons: 1) training on MAP is most commonly used in IR to improve retrieval performance; 2) we find that training on MAP improves the retrieval performance on other metrics as well while training on bpref does not improve the overall performance. Thus, MAP and bpref will both be the primary evaluation measures in this work. In fact, MAP correlates well with bpref as we will show in the next section.

To access the statistical significance of differences in the performance of two systems, we perform one-tailed paired t-test for MAP (since we train systems on MAP).

5. EXPERIMENTS AND RESULTS

This section describes experiments, presents the evaluation results, and discusses the research findings.

5.1 Impact of ICD and NEG

In Section 3.1.1, we introduced feature ICD to incorporate extra information, and NEG to exclude negated information. Now we demonstrate their impact on the retrieval performance of our system. For the baseline system, we use the raw medical corpus and the merge-before-retrieval strategy. Table 2 shows that both ICD and NEG slightly improve MAP scores. This is because ICD helps discover more true positives while NEG helps remove false positives. When combined together, they give a relatively larger improvement.

Since ICD and NEG improve the retrieval performance, our systems will use ICD and NEG by default in the rest of paper. Thus, system BL1+ICD+NEG will be the new baseline system (denoted as *baseline2*) that will be used for the rest of paper.

System	MAP	bpref	P10	Rprec
baseline1 (BL1)	0.386	0.505	0.571	0.403
BL1+ICD	0.409	0.537	0.5647	0.417
BL1+NEG	0.394	0.522	0.5853	0.417
BL1+ICD+NEG	0.416	0.551	0.594	0.434

Table 2: Impact on ICD and NEG. ICD helps discover more true positives while NEG helps remove false positives. Together, they improve the overall performance of the system. System BL1+ICD+NEG also corresponds to baseline2 in Table 6.

5.2 Retrieval before Merging

As mentioned in Section 3.1.2, we have several options for choosing the score merging function f_{RbM} in Equation 2 (i.e., "merging I" in Figure 1) for RbM. Now we describe them formally below:

MAX:

$$\text{score}_{\text{RbM}}(V,Q) = \max(\{\text{score}(r_j^V, Q)\})$$

SUM:

$$\text{score}_{\text{RbM}}(V,Q) = \sum_j \text{score}(r_j^V, Q)$$

ANZ:

$$\text{score}_{\text{RbM}}(V,Q) = \frac{\sum_j \text{score}(r_j^V, Q)}{|\{\text{score}(r_j^V, Q) \neq 0\}|}$$

where again $\text{score}(r_j^V, Q)$ is the language modeling score of the report r_j^V (associated with visit V) with respect to query Q. ANZ stands for "Averaging over Non-Zeros", meaning we only consider reports containing at least one query term. MAX, SUM, and ANZ are similar to CombMAX, CombSUM, and CombANZ proposed by Fox and Shaw [15]. However, CombMAX, CombSUM and CombANZ were used for merging multiple retrieval runs.

Table 3 shows that MAX is superior to SUM and ANZ. This confirms our assumption that we can rely on the strongest evidence (i.e, the most relevant report) of a visit to estimate the relevance of that visit. Thus, we will use MAX for score merging in RbM by default in this paper.

	MAX (default)	SUM	ANZ
MAP	0.416	0.110	0.317

Table 3: Comparison of score merging methods for RbM.

5.3 Evidence Aggregation

Similarly, we also have several options for choosing the score merging function f_{VRM} in Equation 3 (i.e.," merging III" in Figure 1) for VRM, such as CombMNZ, CombANZ, and CombMAX [15]. In our case, we are only merging two rankings. Thus, these merging methods are specified as follows:

CombMNZ:

$$\text{score}_{\text{VRM}}(V,Q) = N_V \cdot [\text{score}_{\text{RbM}}(V,Q) + \text{score}_{\text{MbR}}(V,Q)]$$

CombSUM:

$$\text{score}_{\text{VRM}}(V,Q) = \text{score}_{\text{RbM}}(V,Q) + \text{score}_{\text{MbR}}(V,Q)$$

CombMAX:

$$\text{score}_{\text{VRM}}(V,Q) = \max(\text{score}_{\text{RbM}}(V,Q), \text{score}_{\text{MbR}}(V,Q))$$

CombANZ:

$$\text{score}_{\text{VRM}}(V,Q) = \frac{\text{score}_{\text{RbM}}(V,Q) + \text{score}_{\text{MbR}}(V,Q)}{N_V}$$

where $\text{score}_{\text{VRM}}(V,Q)$ is the merged score for visit V, and $\text{score}_{\text{RbM}}(V,Q)$ and $\text{score}_{\text{MbR}}(V,Q)$ are the scores for V in two different visit rankings as demonstrated in Figure 1, and N_V is the number of rankings that have V in the top 1000 retrieved visits. Note that $\text{score}_{\text{MbR/RbM}}(V,Q) = 0$ if V does not appear in the top 1000 retrieved. We compare the performance of these merging methods using the primary evaluation measures in Table 4. As we can see, CombMNZ and CombSUM achieve comparable performance, and are better than CombMAX and CombANZ. Thus, we can infer that a good aggregation strategy for "merge III" should favor visits that appear in both rankings. We use CombSUM as the default merging method for VRM.

Next, we compare the three evidence aggregation strategies as described in Section 3.1.4. Table 5 shows that VRM

Method	MAP	bpref
CombMNZ	0.446	0.564
CombSUM (default)	0.446	0.563
CombMAX	0.427	0.559
CombANZ	0.356	0.510

Table 4: Comparison of score merging methods for VRM.

is significantly better than MbR and RbM on MAP, which means that merging visit rankings as the top-level evidence aggregation strategy boosts the retrieval performance significantly.

System	MAP	bpref	P10	Rprec
MbR	0.393	0.530	0.565	0.403
RbM	0.416	0.551	0.594	0.434
VRM	$0.446^{\triangle}$	0.563	0.635	0.456

Table 5: Comparison of evidence aggregation methods. $^{\triangle}$ indicates that the MAP difference between VRM and MbR/RbM is statistically significant ($p < 0.05$). System RbM is the same as BL1+ICD+NEG in Table 2.

6. FURTHER IMPROVEMENT

In this section, we further improve our retrieval system by using advanced retrieval models and combining them in an innovative way. We also compare our system with top-performing TREC systems.

6.1 Advanced Retrieval Models

The retrieval model descried in Section 3 is a strong baseline, but the only information it uses is terms in the query and terms in the document. In addition, it assumes independence between query terms. In the following, we introduce several advanced retrieval models to further improve the retrieval performance of our system.

6.1.1 Markov Random Field Model

Medical queries usually contain phases that describe conditions, symptoms, drug names, treatments, etc. These query terms are likely to occur in close proximity to each other in relevant documents. Thus, we use the Markov random field (MRF) model proposed by Metzler and Croft [12] to model term dependencies. MRF effectively expands the original query with phrases. For example, the query "ductal carcinoma in situ" would be expanded with phrases "ductal carcinoma" and "in situ" (but also "carcinoma in"). Each phrase is added as both an ordered phrase and an unordered phrase.

The MRF model has been shown to consistently outperform the standard unigram model across a range of TREC test collections [12, 13]. We use their sequential dependence model in particular. Following Metzler and Croft [12], we set the feature weights $(\lambda_T, \lambda_O, \lambda_U)$ to (0.8, 0.1, 0.1). Here λ_T is the weight given to the original bag-of-words query, λ_O the weight given to ordered phrases, and λ_U the weight given to unordered phrases.

Thus, our first improved system will use VRM as evidence aggregation strategy and use MRF as retrieval model (in place of the baseline retrieval model). We denote this system as VRM-MRF.

6.1.2 Mixture of Relevance Models

Queries specified by the search users can have a "vocabulary mismatch" with the content in a medical report since there are many different ways to express a medical concept (e.g., "hearing loss", "hearing impairment", "difficult of hearing", and even "deafness" are all semantically related, but they only have one common term at the most). The consequence is that the system may have a relatively low recall if there is a "vocabulary mismatch". We can alleviate this problem and improve our baseline retrieval model by expanding the query with additional "related" terms. These related terms (also called expansion terms) can be derived from a relevance model θ_Q, which is usually built upon top-ranked k documents for the query in the target collection (i.e., the same collection used for retrieval).

Thus, in this paper we derive expansion terms based on their weights p which are estimated by:

$$p_i = \sum_{j=1}^{k} \exp\{\frac{tf_{e_i,D_j}}{|D_j|} + \log\frac{|C|}{df_{e_i,C}} + score(D_j, Q)\}, \quad (6)$$

where $score(D_j, Q)$ is the query likelihood score for the top jth feedback document in the initial retrieval set ranked by Equation 1, tf_{e_i,D_j} is the term frequency of e_i in document D_j, $df_{e_i,C}$ is the document frequency of e_i in collection C, and $|D_j|$ and $|C|$ are document and collection lengths in words respectively. This formula estimates the importance of term e_i based on its term frequency, inverse document frequency, and feedback document scores. m terms with highest scores p are selected as expansion terms, and they form our estimated relevance model $\hat{\theta}_Q$. Note that we also normalize p so that we have an estimated probability $P(w|\hat{\theta}_Q)$ for each word w.

Relevance modeling can be further improved upon by leveraging information in other document collections. Specifically, following Diaz and Metzler [4], we can form relevance models for two or more additional collections, then expand the query using those models.

To achieve better performance, we linearly interpolate the mixture of relevance models with the maximum likelihood (ML) query estimate by formulating the equation:

$$P(w|\theta_Q) = \lambda_Q \frac{\#(w,Q)}{|Q|} + \sum_C \lambda_C P(w|\hat{\theta}_{Q,C}), \quad (7)$$

where the first part is the weighted ML query estimate for word w and the second part represents the mixture of relevance models. In particular, $P(w|\hat{\theta}_{Q,C})$ is the probability of w in the estimated relevance model $\hat{\theta}$ built upon top-ranked documents in expansion collection C. λ's are collection weights and $\lambda_Q + \sum_C \lambda_C = 1$. Indri naturally supports such queries with the "#weight" operator; we implement Equation 7 in Indri by formulating a query of the following format:

#weight(
  λ_Q #combine(w_1 w_2 ... $w_{|Q|}$)
  λ_{C_1}#weight(p_{11} e_{11} p_{12} e_{12} ... p_{1m} e_{1m})
  ...
  λ_{C_n}#weight(p_{n1} e_{n1} p_{n2} e_{n2} ... p_{nm} e_{nm})
).

Here w_i represents a term in the original user query; e_{ij} represents the jth expansion term (in decreasing order of

probability p_{ij}) from collection i, n is the number of expansion collections, and m is the number of terms to expand with. The "#combine(w_1 w_2 ... $w_{|Q|}$)" phrase corresponds to the ML query estimate while the "#weight(p_{i1} e_{i1} p_{i2} e_{i2} ... p_{im} e_{im})" phrase corresponds to the estimate of relevance model $\hat{\theta}_{Q,C_i}$. Note that p will be automatically normalized by the "#weight" operator in Indri.

Thus, an expanded query based on two expansion collections when the values of λ's are specified as (0.7, 0.2, 0.1) looks like the following:

```
#weight(
  0.7 #combine( female breast cancer mastectomies admission )
  0.2 #weight( 0.225 mastectomy 0.145 women 0.110 risk
      0.107 prophylactic 0.101 bct 0.074 radiate 0.068 therapy
      0.062 radiotherapy 0.058 surgery 0.050 adjuvant )
  0.1 #weight( 0.211 mammographic 0.159 tram 0.101 dci
      0.116 mammography 0.93 flap 0.082 mammogram
      0.068 duct 0.063 biopsy 0.059 axillary 0.048 recurrence )
).
```

For this work, we use three expansion collections. The first is the medical records corpus itself, the other two are TREC 2007 Genomics Track dataset [9] and TREC 2009 ClueWeb09 Category B dataset[7]. The ClueWeb09 dataset was created to support research on information retrieval and related human language technologies. It consists of about 1 billion web pages in ten languages that were collected in January and February 2009. The dataset is used by several tracks of the TREC conference. TREC Category B contains first 50 million English pages[8]. We choose these collections because there are existing topics and relevance judgments for analysis. As for model parameters, we set λ's to (0.7, 0.1, 0.1, 0.1) and use top 10 terms from top 50 feedback documents.

Thus, our second improved system will use VRM as evidence aggregation strategy and use Mixture-of-Relevance-Models (MRM) as the retrieval model (in place of the baseline retrieval model). We denote this system as VRM-MRM.

6.1.3 A Combined Model

We linearly combine MRF and MRM to get our third retrieval model. The scoring function looks like

$$P(w|\theta_Q) = \lambda_Q \cdot \text{MRF} + \sum_C \lambda_C P(w|\hat{\theta}_{Q,C}), \qquad (8)$$

which is similar to Equation 7. The difference is that we replace the ML query estimate with MRF. This is an novel way to combine two proven statistical IR models. The new retrieval model (denoted as MRF-MRM) is expected to benefit from term dependence modeling as well as query expansion.

Thus, our final system will use VRM as evidence aggregation strategy and MRF-MRM as retrieval model. We denote this system as VRM-MRF-MRM.

6.2 Comparing with TREC systems

We compare our improved systems with three representative TREC systems in Table 6. Among TREC automatic systems, CengageM11R3 [10] was ranked 1st overall and 1st in the judged group while UTDHLTCIR [5] was ranked 3rd overall but 1st in the unjudged group[9] [16]. These two systems both benefited from ICD code expansion, negation removal, and age/gender filtering. Moreover, CengageM11R3 was trained on a set of in-house-built relevance judgments with 30 topics that are similar to TREC official topics. Their training results agreed with the TREC results in terms of ranking their four system variants [10]. NLMManual [3] was the only TREC manual system that outperformed UTDHLTCIR. Medical experts were allowed to look at the retrieving results and modify the queries accordingly in the manual systems.

System	MAP	bpref	P10	Rprec
NLMManual	0.507	0.658	0.727	0.500
VRM-MRF-MRM*	$0.501_{(+20\%\dagger)}$	0.631	0.656	0.505
VRM-MRM*	$0.475_{(+14\%\dagger)}$	0.611	0.632	0.481
VRM-MRF*	$0.468_{(+13\%\dagger)}$	0.585	0.644	0.486
VRM*	$0.446_{(+7\%\dagger)}$	0.563	0.635	0.456
CengageM11R3	0.457	0.552	0.656	0.440
baseline2* (RbM)	$0.416_{\dagger}$	0.551	0.594	0.434
UTDHLTCIR*	0.408	0.545	0.603	0.422

Table 6: Performance comparison. Systems are ordered by bpref as in [16]. TREC systems are in gray background. Systems starred did not contribute to the TREC judgment pools and thus the scores are the lower bounds for those systems. The †'s indicate the improvement over baseline2 (Note that basline2 is same as RbM in Table 5 and BL1+ICD+NEG in Table 2). The MAP differences between 1) VRM-MRF and VRM, 2) VRM-MRM and VRM, 3) VRM-MRF-MRM and VRM-MRF, and 4) VRM-MRF-MRM and VRM-MRM, are all statistically significant ($p < 0.05$), meaning each further improvement significantly boosts the retrieval performance. System VRM-MRF-MRM improves baseline2 by 20% on MAP.

Table 6 compares the performance of our systems and TREC systems. The MAP differences between 1) VRM-MRF and VRM, 2) VRM-MRM and VRM, 3) VRM-MRF-MRM and VRM-MRF, and 4) VRM-MRF-MRM and VRM-MRM, are all statistically significant ($p < 0.05$), meaning each further improvement significantly boosts the retrieval performance.

Our best system VRM-MRF-MRM outperforms the best TREC automatic system on MAP, bpref, and Rprec. Though VRM-MRF-MRM does not outperform the best TREC manual system, our results have shown that we have made substantial progress towards one of the ultimate goals of medical records track, which is to build an automatic medical record search system that can be as reliable as a manual search system. Note that because our system did not contribute runs to TREC judgment pools, our scores could potentially be higher.

We further compare the "recall" of the NLMManual and VRM-MRF-MRM in Table 7. As we can see, VRM-MRF-MRM has a slightly higher recall than NLMManual. The recall is also an important factor in evaluating a medical record search system. For example, if one wants to find a large number of patients matching certain criteria for clini-

[7]Available at http://lemurproject.org/clueweb09.php/

[8]In this work, we excluded all the Wikipedia pages (about 6 million) in Category B

[9]Systems in the judged group contributed runs into the judgment pools while systems in the unjudged group did not.

	NLMManual	VRM-MRF-MRM
# of relevant visits that are retrieved	1609	1646
total # of relevant visits for all topics	1765	
recall	91.2%	93.3%

Table 7: Comparison of recalls.

cal studies so that the outcome can have a large statistical power, a system with a high recall will greatly reduce the manual effort in post-filtering false retrieval hits.

Finally, we highlight our research findings: 1) we have evaluated a series of helpful features, including document pre-processing methods (i.e., ICD and NEG), different evidence aggregation strategies (i.e., RbM, MbM, VRM), and advanced retrieval models (i.e., MRF and MRM), for building a medical records search system. Each feature focuses on a different aspect of the system design and improves the overall system performance. We argue that for search systems to do well in retrieving relevant medical records they need to be built upon similar features. 2) The Genomics and ClueWeb datasets in the MRM model help a lot though they are not clinical in nature, which may suggest the possibility of further improvement with the addition of a collection of documents that better relate to those in the clinical set.

7. RELATED WORK

As EHR become more prevalent, attempts have been made to transfer search engine technology to EHR retrieval for various applications [6]. The EMERSE (Electronic Medical Record Search Engine) system, as one of the earliest and successful non-commercial EHR search engines, has been used by medical professionals in a few hospitals, health centers, and clinics since its initial introduction in 2005 [6, 14]. EMERSE supports free-text queries and offers several advanced features such as query suggestion and collaborative search [18]. Though EMERSE has not achieved widespread adoption and there is little discussion about its search algorithms, a few interesting research work have been done using the EMERSE system:

Seyfried et al. [14] compared EMERSE-facilitated chart reviews with manual reviews, and concluded that using a well-designed EHR search engine for retrieving information in free-text EHR can provide significant time saving while preserving reliability.

Yang et al. [17] analyzed a query log of the EMERSE system recorded over the course of 4 years. One of their interesting findings is that the coverage of EHR query terms by a meta-dictionary (containing all terms in Unified Medical Language System, an English dictionary, and a medical dictionary) is much lower than the usual 85-90% coverage of Web queries by English dictionaries. Thus, they suggested seeking beyond the use of medical ontologies to enhance medical information retrieval. This can actually explain why our MRM model works so well.

Apart from these few attempts, methods emerging from research on information retrieval have not been well explored, largely due to the sensitivity of patient data, preventing its use by academic researchers. Fortunately, The Text REtrieval Conference (TREC) initiated a medical records track in 2011 making a set of real medical records and human judgments of relevance to search queries available to the research community.

Most TREC participants of medical records track used domain-specific knowledge to enhance retrieval. King et al. [10] annotated segments of the report text as having specific properties/features. They also identified and indexed terms of medical reports that appeared in the Unified Medical Language System (UMLS) Metathesaurus. Meanwhile, they expanded original queries with related terms in UMLS and several commercial medical reference encyclopedias. Their best run improved their baseline by about 18%.

Goodwin et al. [5] used several external utilities for query expansion, such as PubMed Central Open Access Subset (a small portion of PubMed Central database), SNOMED-CT (Systematized Nomenclature of Medicine-clinical Terms), and UMLS. They found that using these external medical-related sources together improved their baseline system performance.

Limsopatham et al. [11] made use of the ICD codes in the reports and enriched reports with ICD code descriptions and related Wikipedia pages. They identified medical concepts in both documents and queries based on medical-domain ontologies in SNOMED-CT and Medical Subject Headings (MeSH), and expanded the concepts with nearby concepts in the ontology hierarchies (i.e., trees in MeSH, ICD, and SNOMED). They also obtained promising results.

The above top-performing TREC systems also used similar ICD expansion and negation detection methods. That is why we included our ICD and NEG features in our system *baseline2* in Table 6.

Overall, our system differs from these TREC systems in three aspects:

1. Evidence aggregation
   TREC systems used approaches similar to either MbR (e.g., [10] and [19]) or RbM (e.g., [11]) for evidence aggregation. However, our system leveraged both strategies and achieved a significant performance improvement.

2. Retrieval Models
   Our system used two proven statistical IR models to tackle two specific problems in search in medical domain, i.e., modeling term dependencies and solving vocabulary mismatch. We combined these retrieval models in an innovative way and boosted the overall performance of the system significantly.

3. Expansion sources
   TREC systems extensively used domain-specific metathesauri for query expansion. However, we leveraged information in general data sources, and effectively extracted useful expansion terms and improved the system performance significantly.

8. CONCLUSION AND FUTURE WORK

In this work, we present a system for searching flat-text health records for patients matching certain inclusion criteria for clinical studies. Our system improves a strong baseline in two aspects. First, it aggregates multi-level evidence in a novel way and significantly boosts its retrieval performance. Second, it innovatively combines two proven statistical IR models to tackle two specific problems of search in medical domain, and further improves the retrieval performance significantly. As our cross-validation results show,

our system achieves a 20% MAP (mean average precision) improvement over a strong baseline (i.e., VRM-MRF-MRM vs. *baseline2* in Table 6). Moreover, the overall performance of our system is comparable to other top-performing systems on the same task.

For future work, we may explore incorporating medical meta-thesari in the MRM model, and compare their performance with general data collections. Another direction would be to design a query-adaptive expansion strategy, meaning that the system would expand each specific query using only the most relevant expansion collections instead of using all of them.

9. ACKNOWLEDGEMENTS

The authors would like to thank the University of Pittsburgh and NIST for providing the medical corpus.

10. REFERENCES

[1] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. Evaluation of negation phrases in narrative clinical reports. *Proceedings of AMIA Symposium*, pages 105–109, Jan. 2001.

[2] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 1 edition, Feb. 2009.

[3] D. Demner-Fushman, S. Abhyankar, A. Jimeno-Yepes, R. Loane, B. Rance, F. Lang, N. Ide, E. Apostolova, and A. R. Aronson. A knowledge-based approach to medical records retrieval. In *Proceedings of The 20th Text REtrieval Conference*, 2011.

[4] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2006. ACM.

[5] T. Goodwin, B. Rink, K. Roberts, S. M. Harabagiu, and R. Tx. Cohort shepherd: Discovering cohort traits from hospital visits. In *Proceedings of The 20th Text REtrieval Conference*, 2011.

[6] D. A. Hanauer. EMERSE: The electronic medical record search engine. *AMIA Annual Symposium Proceedings*, 331(7531):941, Jan. 2006.

[7] H. Harkema, J. N. Dowling, T. Thornblade, and W. W. Chapman. Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839–851, 2009.

[8] W. Hersh. *Information Retrieval: A Health and Biomedical Perspective*. Health Informatics. Springer, third edition, 2009.

[9] W. R. Hersh, A. M. Cohen, L. Ruslen, and P. M. Roberts. TREC 2007 genomics track overview. In *TREC*, 2007.

[10] B. King, L. Wang, I. Provalov, and J. Zhou. Cengage Learning at TREC 2011 medical track. In *Proceedings of The 20th Text REtrieval Conference*, 2011.

[11] N. Limsopatham, C. Macdonald, I. Ounis, G. Mcdonald, and M. Bouamrane. University of Glasgow at medical records track 2011: Experiments with Terrier. In *Proceedings of The 20th Text REtrieval Conference*, 2011.

[12] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, page 472, 2005.

[13] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10:257–274, June 2007.

[14] L. Seyfried, D. Hanauer, and D. Nease. Enhanced identification of eligibility for depression research using an electronic medical record search engine. *International Journal of Medical Informatics*, 78(12):e13–e18, Dec. 2009.

[15] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252, 1994.

[16] E. M. Voorhees and R. M. Tong. Overview of the TREC 2011 medical records track. In *TREC*, 2011.

[17] L. Yang, Q. Mei, K. Zheng, and D. Hanauer. Query log analysis of an electronic health record search engine. In *AMIA Annual Symposium Proceedings*, pages 915–924, 2011.

[18] K. Zheng, Q. Mei, and D. Hanauer. Collaborative search in electronic health records. *Journal of the American Medical Informatics Association*, 18(3):282–291, 2011.

[19] D. Zhu and B. Carterette. Using multiple external collections for query expansion. In *Proceedings of The 20th Text REtrieval Conference*, 2011.