Three Questions about Clinical Information Retrieval

Stephen Wu, James Masanz, Ravikumar K.E., Hongfang Liu
Mayo Clinic, Rochester, MN

1 Introduction

Electronic Medical Records (EMRs) have greatly expanded the potential for the evidence-based improvement of clinical practice by providing a data source for computable medical information. The Text REtrieval Conference 2012 Medical Records Track (TREC-med) explored how information retrieval may support clinical research by providing an efficient means to identify cohorts for clinical studies. A shared task called participants to find cohorts of relevant patients for 50 different topic queries.

The users in TREC-med information retrieval systems would be medical experts who are searching for cohorts. In our previous work, we have collaborated with such experts on specific queries; the assortment of 50 queries makes this competition a standardized benchmark task. Thus, techniques that have shown case-by-case improvement can be tested against a much larger number of queries. We have taken this opportunity to investigate three core questions around which many of our algorithms are designed:

1. What is the relative value of structured data (e.g., fields in EMRs, or document metadata) compared to clinical text?
2. Are extensive information extraction (IE) efforts any benefit when we consider the applied question of information retrieval (IR)?
3. Can distributional semantics help supply missing information in a query?

For each of these three questions, we have extended Apache Lucene[1] with pre-existing techniques and tested on the TREC-med cohort identification task. In testing these independently, we aim to find generalizable principles for cohort identification in other documents collections and queries.

The rest of this paper describes the TREC 2012 Medical Records task, describes Mayo Clinic's run submissions in detail, and reports evaluation results with subsequent discussion.

[1]See `lucene.apache.org`

2 Background

The TREC 2012 Medical Records track was arranged as a follow-up to the 2011 track [1], with nearly identical setup. The data to be retrieved lay in the University of Pittsburgh's BLU repository, which includes the free text portions of medical records (see `report_text` below). Each patient at the University of Pittsburgh would have one or more medical *records* (documents) associated with him or herself. Each record was given in XML format, and included both structured data and the unstructured text.

```
<?xml version=''1.0'' encoding=''UTF-8''
standalone=''no''?>
<report>
<checksum>20060201ER-Fs2xiJYPXwVE-848-1341620775
</checksum>
<subtype>EVAL</subtype>
<type>ER</type>
<chief_complaint>DENTAL
PAIN</chief_complaint>
<admit_diagnosis>521.00</admit_diagnosis>
<discharge_diagnosis>525.9,E917.9,
</discharge_diagnosis>
<year>2007</year>
<downlaod_time>2008-02-06</downlaod_time>
<update_time/>
<deid>v.6.22.06.0</deid>
<report_text>[Report de-identified
(Safe-harbor compliant) by De-ID
v.6.22.06.0]
.
.
.
</report_text> </report>
```

Records are uniquely identified by their `checksum`. Note that each record contains a note `type` and `subtype`; in the example, the note comes from an Emergency Room/Department. The `chief_complaint` section is a helpful textual summary of what the record is about from the patient's perspective, but is not present for every record. The `admit_diagnosis` and `discharge_diagnosis` serve

a similar function but are also not always present. They are given as ICD-9 codes, a medical terminology frequently used for billing purposes. Finally, notice that the notes were de-identified, so that any protected health information has been replaced with surrogates.

The records were grouped into *visits* — a physical visit to the hospital. The unit of retrieval was defined as a patient visit. In total, there were 95,702 records that corresponded to 17,198 visits. The largest visit was 418 records, but the mean visit was 5.56 records.

Participants from 24 institutions were given a set of 50 hypothetical topics (queries) developed by experts at the Oregon Health Sciences University (OHSU). Each topic query is given in a form such as

> Number: 143
> Patients who have had a carotid endarterectomy

These topics defined patient profiles that might be involved in a clinical trial. For each topic, participants retrieved a list of patient visits in order of relevance to the topic.

For evaluation and ranking, retrieved records from participants' runs were given to assessors at OHSU. These assessors rendered relevance judgments on a stratified pool of visits — the top 15 of each submitted run, and a random sample of the remaining top 100 in each run. The nature of each topic and its correspondence with the given dataset varied greatly. For example, 4 topics were discarded for purposes of evaluation because no records were assessed as being relevant to the query topic; on the other hand, other topics likely had many relevant visits that were never assessed.

3 Methods

We tested 3 focused questions that lay groundwork for future patient identification systems. We evaluated the usefulness of document metadata, contextually-aware information extraction results, and distributional semantic query expansion. As a baseline, we used a standard Lucene index, and each of the other runs was built directly on this baseline.

3.1 Baseline

Apache Lucene is perhaps the most widely used information retrieval framework. Lucene indexes a collection of documents for extremely efficient text search. Similar to rows in a database, documents are the granularity of a collection in Lucene; similar to columns, *fields* in Lucene contain values for each document that are considered strings by default. We took each medical record from the BLU repository and stored fields corresponding to 10 useful parts of the XML document: Document ID (checksum), Visit ID (stands in for Patient ID), Date (from checksum), Type, Subtype, Chief Complaint, Admit diagnosis (ICD-9 codes), Discharge diagnosis (ICD-9 codes), Year, and Content (the clinical text).

As mentioned, each of the fields in Lucene can be indexed. Behind the scenes, by default, Lucene uses an Analyzer pipeline for the text that includes tokenization, normalization, lowercasing words, and stop word removal; it then creates an *inverted index* for each token in each field. Unlike the mapping implied in the fields (from documents to tokens), an inverted index maps from tokens to documents, and this makes it easy to find terms that match a query. For our baseline methodology, we included variants of "patient" as a stop word, and only searched the Content field.

When searching for a term, Lucene effectively uses the following equation to rank which documents are most relevant:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{qNorm}(q) \cdot \sum_{t \in q} \left(\text{tf}(t) \cdot \text{idf}(t)^2 \cdot \text{t.boost}() \cdot \text{norm}(t,d) \right) \quad (1)$$

where q is the query, d is a document in the collection, t is a term, and the following functions hold:

score	the document's score
coord	weight for # term matches btwn. q & d
qNorm	normalizes similarities between queries
tf	square root of term frequency
idf	$1 + \log(D/(\text{df}(t)+1))$, df() is doc. freq.
t.boost	weight for query terms
norm	weights for fields & terms in a document

In this baseline approach, we strictly used the text of the original TREC topic as the query, without any special weighting for t.boost(). It should be noted, however, that we used Lucene's same Standard Analyzer on both the query and the collection. Likewise, we searched within an unaltered Content field (i.e., the text itself), and thus norm() had no effect on the final weighting.

Since Lucene ranks top *documents* rather than top *visits* (i.e., patients), we consider the most relevant document to represent the whole visit. This maximum-document assumption for each patient is not neceessarily a good one, but we have focused on testing other aspects of the retrieval framework. We report the top 1,000 *unique* visits for each query.

Evaluations in TREC-med 2011 were on a smaller set of 35 topics, and reported results used bpref; the performance on TREC-med 2011 topics was bpref=0.4249.

3.2 Test 1: Including Structured Metadata (MayoMetadata)

First, we performed a controlled test of the value of EMR structured data on retrieval. For practitioners and researchers who use EMR data, ICD-9 codes are the first line of defense in cohort identification. They are frequently used in lieu of more sophisticated cohort identification procedures. We accounted for these ICD-9 codes by mapping to their textual representations using the UMLS Metathesaurus. Term lists for each ICD-9 code were then stored in additional Lucene index fields: Admit Diagnosis Terms and Discharge Diagnosis Terms.

This is not strictly the same as a boolean search for matching ICD-9 codes, which requires person or process to code the query into ICD-9 codes. However, in our experience, users of an IR system will typically think of terms of interest, then consult a reference material to find relevant codes, then search the structured data for codes. Indexing and searching the textual representations of ICD-9 codes is thus a reasonable automatic method for retrieving ICD-9 codes.

With the query unchanged from the baseline, we searched over 4 fields: the text of these two diagnoses terms, the Chief Complaint, and the Content field. Since we expected these coded representations to be highly relevant for retrieval, we weighted the structured data sections higher (in proportion to the much shorter field length) than the text itself, by using the norm() function. Performance on the 35 TREC 2011 topics improved significantly using this addition (bpref=0.4541).

3.3 Test 2: Weighting IE Output (MayoPayload)

Second, we performed a controlled test (i.e., ignoring metadata) of the value of using the results of information extraction to inform the scoring procedures. We used a recent Information Extraction (IE) system developed at Mayo Clinic, MedTagger, due to its speed. MedTagger uses a lexicon of terms and variants that have been attested in a large corpus of clinical text [2, 3], and looks for these terms in the document collection (here, the BLU NLP repository) as its means of Named Entity Recognition. Additionally, MedTagger applies the NegEx [4] and ConText [5] algorithms, which discover whether these named entities were negated, hypothetical, historical, or experienced by someone other than the patient.

After being found in the BLU text, these named entities and their attributes were brought into the Lucene index. For each named entity, the last token in that named entity carried a *Payload* — additional data attached to a token within Lucene. Our custom-defined payload included a normalized form of the named entity, the semantic group, status (any hedging of a statement), polarity (whether the statement was negated), and the experiencer (subject of a statement, typically the patient).

We used a simple heuristic to down-weight tokens if its attributes cast any doubt that the named entity was associated with the patient. Values were chosen by manually testing against the 2011 query topics:

$$\begin{aligned} \textbf{polarity} &= \text{“negated”} \rightarrow .10w \\ \textbf{status} &= \text{“history of”} \rightarrow .75w \\ &\quad \text{“fam. history”} \rightarrow .10w \\ &\quad \text{“probable”} \rightarrow .25w \\ \textbf{experiencer} &= \text{not “patient”} \rightarrow .10w \end{aligned} \tag{2}$$

This weight w is an incremental part of the calculation for $\text{norm}(t, d)$.

By augmenting the index with the ability to downweight on these mentions, query terms finding negated or hedged matches would be dispreferred. This simple NLP-driven addition improved performance from the baseline on TREC-med 2011 topics, with bpref=0.4730.

3.4 Test 3: Query Expansion with Semantic Vectors (MayoExpanded)

Finally, we performed a controlled test (i.e., ignoring metadata and negation) of the value of query expansion through distributional semantics. While the baseline and previous approaches directly used the text of the queries (with stop word removal) to search documents, here we modified the queries.

We used Random Indexing [6] to build distributional semantic representations (i.e., vectors) of terms from a large corpus of Mayo Clinic clinical notes. Near-neighbor terms (often multiple tokens each) were selected for each topic. We constructed expanded queries whose terms had a t.boost() weighting based on the frequency of tokens in the near-neighbor synonymous term lists. We included it for diversity in the result pool, despite slightly decreased performance on TREC 2011 topics (bpref=0.4097).

	infNDCG	$\sigma^2_{\text{infNDCG}}$	infAP	bpref	R-prec	P@10
MayoLucene	0.3694	0.2310	0.1359	0.2771	0.2583	0.4043
MayoMetaData	0.3222	0.2125	0.1175	0.2474	0.2072	0.3553
MayoPayload	0.4119	0.0634	0.1590	0.2981	0.2807	0.4319
MayoExpanded	0.3587	0.1422	0.1239	0.2652	0.2321	0.4043

Table 1: MayoClinicNLP results for TREC-med 2012

	2011 topics		2012 topics	
	bpref	Δ	bpref	Δ
MayoLucene	0.4249	–	0.2771	–
MayoMetaData	0.4541	+6.87%	0.2474	-10.72%
MayoPayload	0.4730	+11.32%	0.2981	+7.58%
MayoExpanded	0.4097	-3.58%	0.2652	-4.29%

Table 2: Comparison of methods on 2011 topics vs. 2012 topics.

	Lucene	MetaData	Payload	Expanded
Lucene	–	0.6451	0.9632	0.8125
MetaData	0.6451	–	0.6336	0.6279
Payload	0.9632	0.6336	–	0.7770
Expanded	0.8125	0.6279	0.7770	–

Table 3: Correlations between the 2012 runs.

4 Evaluation

Official TREC results on the baseline and variants are shown in Table 1. In 2011, bpref was used for the official evaluation rankings, due to problems in calculating the inferred measures. The inferred measures [7] are now available in 2012.

The baseline (MayoLucene) is improved upon by the IE-influenced retrieval (MayoPayload) across all metrics. Both ICD-9 codes (MayoMetaData) and semantic vector query expansion (MayoExpanded) actually decrease the performance across all metrics. For comparison, Table 2 shows the performance of these techniques on both the 35 topics from 2011 and the 50 topics from 2012. The Δ columns highlight the % difference from baseline associated with each of the three techniques being tested.

Finally, in Table 3 we show the correlations between our baseline and 3 other runs.

5 Discussion

At the beginning, we set out to answer three questions, which we revisit here.

5.1 Structured clinical data in IR? (MayoMetaData)

The evaluation results for 2012 topics suggest that structured data does not uniformly improve performance. This is especially interesting given that performance did improve significantly for 2011 topics when structured data was added, and given that the median R-precision scores for 2011 and 2012 were very similar.

In 2012, 32 of the 47 topics (68%) were hurt by including ICD-9 codes, showing that the detrimental effects of the structured data were relatively widespread. MayoMetaData had the highest standard deviation (0.2125) among the three tested systems.

All this may indicate that topics vary widely, and structured data is not always a good match for what an end user is looking for. Alternatively, a possible explanation is that the 2012 topics are harder to encapsulate in diagnosis codes.

5.2 IE results in IR? (MayoPayload)

Here, we find a clear indication that Information Extraction methods do benefit Information Retrieval. This is shown with positive results across both 2011 and 2012 topic sets.

For 2012 topics, 35 of 47 topics (74%) were im-

proved by including this type of contextual information. This is an encouraging result, showing that extensive IE research has practical benefit in IR systems. MayoPayload also has the smallest standard deviation (0.0634), showing the consistency of the approach.

5.3 Distributional semantic query expansion? (MayoExpanded)

The flavor of query expansion that we have proposed shows an overall drop in performance, showing that query expansion through distributional means cannot necessarily be relied upon. There are indeed cases where query expansion outperforms plain queries (25 of 47, or 53% of cases), but the opposite is also true (22 of 47, or 47%). One contributing factor could be that the semantic vectors were trained on a different distribution of data (Mayo EDT) than the test data (Pittsburgh BLU). Since most TREC 2012 topics had few relevant visits and the goal of query expansion is to aid in increasing recall, it seems that increases in performance due to recall are more than balanced out by the cost in precision.

5.4 Other questions

As shown in Table 3, the correlative relationships between the different approaches confirm existing findings. Overall, MayoLucene is a strong baseline, and large deviations from it tended to come from lower-performing runs. In particular, Because MayoMetaData searches (and highly weights) fields that are not present in the other approaches, it is the least correlated with the others.

Two related, untested questions are whether interactive user input and structured (or faceted) queries would be of benefit. This is part of our future work.

6 Conclusion

The TREC 2012 Medical Records track competition provided an opportunity to test three focused questions about Information Retrieval in the clinical domain. Structured data and query expansion can sometimes be helpful, but information extraction results can be used effectively to greatly increase IR performance.

References

[1] E. Voorhees and R. Tong. Overview of the trec 2011 medical records track. In *The Twentieth Text REtrieval Conference Proceedings TREC*, 2011.

[2] Stephen Wu and Hongfang Liu. Semantic Characteristics of NLP-extracted Concepts in Clinical Notes vs. Biomedical Literature. In *Proceedings of AMIA 2011*, 2011.

[3] Stephen Wu, Hongfang Liu, Dingcheng Li, Cui Tao, Mark Musen, Christopher Chute, and Nigam Shah. UMLS Term Occurrences in Clinical Notes: A Large-scale Corpus Analysis. In *Proceedings of the AMIA Joint Summit on Clinical Research Informatics*, 2012.

[4] W.W. Chapman, W. Bridewell, P. Hanbury, G.F. Cooper, and B.G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301–310, 2001.

[5] 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.

[6] D. Widdows and K. Ferraro. Semantic vectors: a scalable open source package and online technology management application. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 1183–1190, 2008.

[7] E. Yilmaz, E. Kanoulas, and J.A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610. ACM, 2008.