Identifying Patients for Clinical Studies from Electronic Health Records: TREC Medical Records Track at OHSU

Steven Bedrick[1], Kyle Ambert[2], Aaron Cohen[2], William Hersh[2]
1: Center for Spoken Language Understanding, Department of Biomedical Engineering;
2: Department of Medical Informatics and Clinical Epidemiology
Oregon Health and Science University, Portland, OR

1. Introduction

The task of the TREC 2011 Medical Records Track consisted of searching electronic health record (EHR) documents in order to identify patients matching a set of clinical criteria, a use case that might be part of the preparation of a quality report or to develop a cohort for a clinical trial. The task's various topics each represented a different case definition, with the topics varying widely in terms of detail and linguistic complexity. This use case is one of a larger group that represent the "secondary use" of data in EHRs [1] that facilitate clinical research, quality improvement, and other aspects of a health care system that can "learn" from its data and outcomes [2]. It is made possible by the large US government investment in EHR adoption that has occurred since 2009 [3].

The corpus for this task consisted of a set of 93,552 patient encounter files extracted from an EHR system. Each encounter file represented a note entered by a clinician or a report in the course of caring for a patient. Each note or report was categorized by type (e.g., History & Physical, Surgical Pathology Report, Radiology Report) or in some cases sub-type (e.g., Angiography).

The encounter files were each associated with one of 17,265 unique patient visits to the hospital or emergency department. Most visits (≈70%) included five or fewer encounters; virtually all (≈97%) included less than 20. The maximum number of encounters comprising any visit was 415. Each encounter within a visit shared a short (truncated to 40 characters) "chief complaint" as well as a single "admission" ICD-9 code and a set of "discharge" ICD-9 codes. The number of discharge ICD-9 codes varied widely from visit to visit; the median number of codes per visit was 5, while the maximum was 25. (Note, however, that the number of visits with 25 discharge codes was more than four times greater than the number of visits with 24 codes; for this reason, we suspect that the "true" maximum number of codes per visit may have been higher in the original EHR that gave rise to the corpus, and that the apparent limit of 25 codes per visit is simply an artifact of the export process, much in the same way as the chief complaint field's length was truncated to 40 characters.)

Patients could not be linked across visits, i.e., be identified as having more than one visit, due to the de-identification process applied to the corpus. As such, for the purposes of this task, the "unit of retrieval" was the visit rather than the patient, meaning that the participating systems were to produce a set of matching visits for each topic. Visits could not be tied to individual patients and therefore visit was used as a surrogate for an individual patient meeting the given clinical criteria.

The OHSU team decided to take a manual, interactive approach to the task, and focused on the construction of a search interface that would allow its users to rapidly formulate queries, review their results, and iterate. Using our system, users could search through the TREC 2011 Medical Records Track corpus by any of the various fields (chief complaint, report text, etc.) using a robust search syntax,

and could also include ICD-9 codes in their queries. This allowed for the easy construction of queries representing complex Boolean criteria.

2. Methods

Our system consisted of three main components. The first was a relational database that stored the encounter and visit data, the second was a full-text index of those data, and the third was a Web-based interface that allowed users to search the indexed records. The database component of our system used PostgreSQL, an open-source relational database management system (RDBMS). Using standard relational modeling techniques, we stored the encounter files themselves, as well as their corresponding visits, in a set of database tables. We also included tables to represent admission and discharge ICD-9 codes, as well as their relationships to encounters and visits.

After loading the corpus into the RDBMS, we constructed a full-text index of the corpus' contents using Ferret, which is a complete port of the popular Lucene information retrieval toolkit to the Ruby programming language. Lucene indices are made up of *documents*, each of which contains one or more *fields* consisting of a block of text. For our index, the documents were individual encounters, and the fields were the report text, chief complaint, and admission/discharge ICD-9 codes.

Lucene supports a rich search syntax that allows users to express complex Boolean queries using arbitrary combinations of words, literal phrases, wild-card terms (e.g., `monkey*` to match `monkeys` as well as `monkeying`, etc.), and edit-distance (a.k.a., "fuzzy" searches) terms. Using parentheses and Boolean operators, groups of terms can be combined (e.g., `(carpal | hand) & fracture`), and query terms can be limited to specific index fields. For example, in the query `chief_complaint:mva & report_text:'carpal fracture'`, we are specifying that matching documents' chief complaint fields must both contain the "word" `mva` (abbreviation for `motor vehicle accident`) and have the exact phrase `carpal fracture` in their report text fields.

We wanted users of our system to be able to include ICD-9 codes in search queries, and to do so in ways that could take advantage of wildcards, fuzzy operators, etc. For example, a user might wish to search for records whose discharge ICD-9 codes involved sprains by using a wildcard operator: `discharge_icd_code:84*`; if they were interested in only sprains of the ankle and foot, they could instead search for `discharge_icd_code:845*`. Since each record contained several discharge ICD-9 codes, and Lucene's index format only supports single-string fields, we concatenated the ICD-9 code numbers using white space to produce the discharge ICD-9 code field. So, for example, an encounter with discharge ICD-9 codes of 881.00, V06.5, and E920.8 would be entered into our index as `881.00 V06.5 E920.8`.

Lucene's standard indexing algorithm tokenizes fields using punctuation and white space, and generally ignores tokens that consist entirely of numbers. While this is appropriate for the general problem of indexing natural text, it is too aggressive for use with many forms of semi-structured text (especially semi-structured text composed primarily of numbers and semantically-meaningful punctuation, such as whitespace-delimited lists of ICD-9 codes).

Fortunately, Lucene's architecture is explicitly intended to be easy to extend in situations such as these. We were able to implement a custom extension to Lucene's tokenization components that dynamically

adjusts the tokenization algorithm on a per-field basis. For the textual fields (chief complaint, report text, etc.), our system uses Lucene's standard tokenizer; for the ICD-9 field, however, it uses a simple whitespace tokenizer that preserves the code numbers in their entirety.

After indexing the corpus, we used the Ruby on Rails Web programming framework to construct a simple user interface to the index and database. Users of the interface can search using either simple keywords or the more complex syntax described previously, depending on their level of expertise and need. Users can also search by admission or discharge ICD-9 code, and the interface includes several features designed to make it easier for users unfamiliar with the ICD-9 system to identify and select code numbers or ranges for inclusion in their queries. The search interface also allows users to limit their results by report type, for example to only retrieve results where the query matches for an emergency room admission note encounter. Since some report types (particularly pathology and radiology reports) tend to be "noisier" than others for some topics, we found this to be a helpful feature.

The interface allows users to view results in a fairly standard Web-style interactive mode, in which it is possible to view related encounters from the same visit, browse encounters by ICD code, and otherwise explore the data in the corpus, all by clicking on appropriate links. In addition to this mode, the interface allows its users to automatically download a set of search results as a `trec_eval` run file, and even to upload a set of queries in order to automatically generate a multi-topic run file.

Since our runs were fully manual, we used two humans to generate the queries that ultimately produced our run submissions. The first human was one with a clinical background (WH), who reviewed the free-text topic descriptions and "translated" each one into a set of keywords and Boolean operators. The second human (SB) was an informatician familiar with the task corpus, who took the clinician's keyword sets and produced formal Lucene queries using the various syntax features (wildcards, etc.) previously discussed.

Our official runs for the track (submitted before the TREC 2011 conference) consisted of the free-text queries, augmented with ICD-9 codes when the free-text queries retrieved zero results, first against the entire corpus, and then just applied to document types in the corpus we thought might achieve better results. These were the discharge summaries and the emergency room visits (the latter for visits for which there might have been no hospital admission). Thus the official OHSU runs were *ohsuManAll*, in which the system considered all encounter types as potentially relevant, and *ohsuManLim*, in which the system only considered discharge summaries and emergency room notes. In the case of two topics, the "limited mode" failed to retrieve any documents at all; in these cases, the second searcher continued iterating the query design until at least one document was retrieved. The informatician also incorporated ICD-9 codes into the queries, and went through several rounds of iterative query development using the system's interactive search mode.

Our follow-up approach after the TREC 2011 conference was more systematic. Rather than add ICD-9 terms in an ad hoc manner, we developed ICD-9 queries for the 31 topics for which it was possible (i.e., some topics were not amenable ICD-9 queries, such as those not mentioning an explicit diagnosis). We then combined the textual and ICD-9 queries with Boolean AND and OR operators. We ran these three permutations (text only, text OR ICD-9, and text AND ICD-9) against both the complete and the limited (discharge summary and emergency room reports only) document sets.

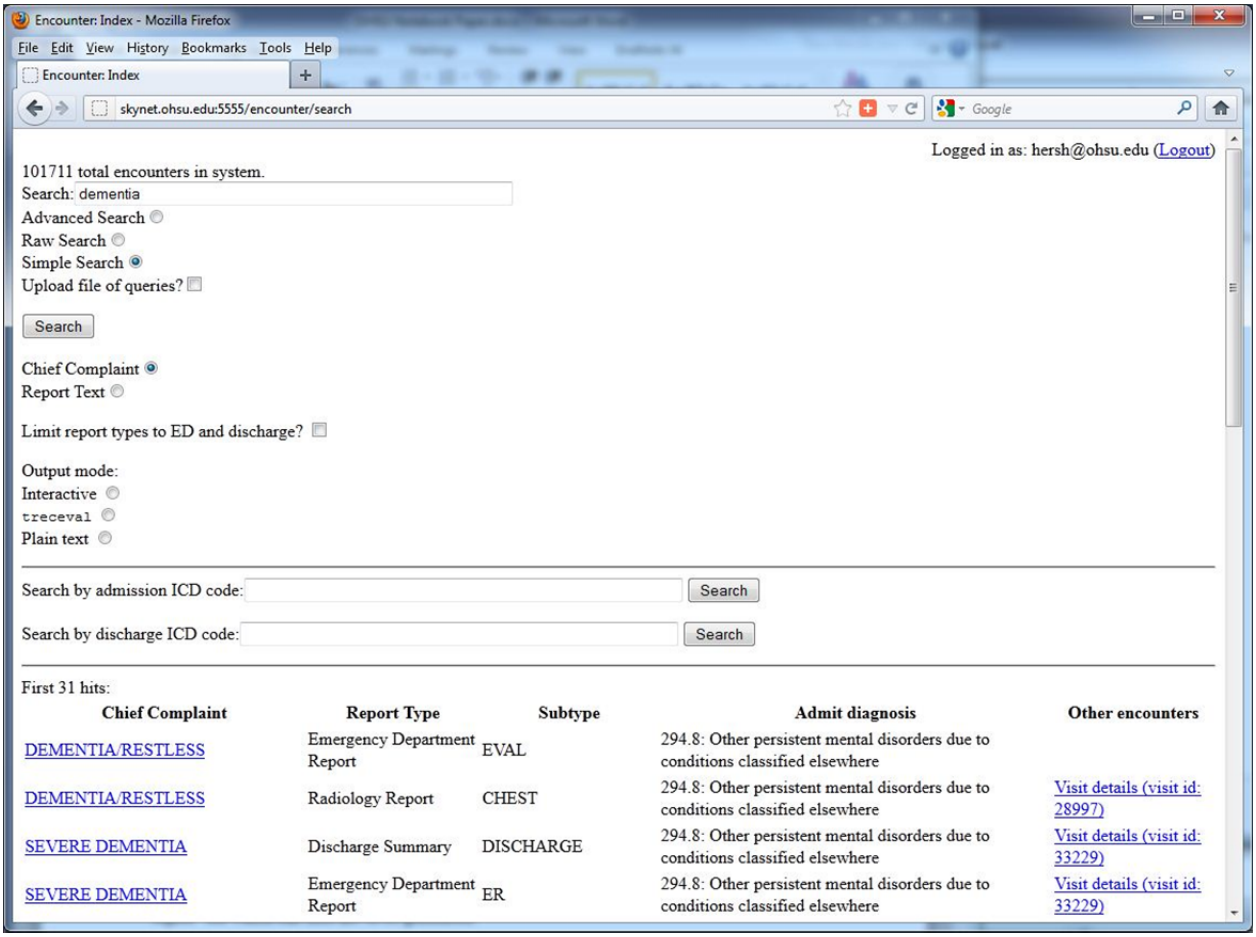

Figure 1 – OHSU TREC Medical Records Track system manual query interface.

3. Results

The two official OHSU runs: ohsuManAll and ohsuManLim, were submitted and are listed in the official NIST results. The six additional runs are listed with the official runs in Table 2.

Our overall results lead to some clear conclusions. First, limiting documents to discharge summaries and emergency room reports reduces overall performance. Second, combining ICD-9 codes with OR slightly decreases overall performance, while combining them with AND significantly decreases it.

As always in information retrieval, averages obscure performance on individual topics. Figure 2 shows the Bpref results of individual topics, which vary widely. Even furthermore, it can be seen that the best-performing condition, Text-only – All, does not perform best on all topics. In fact, as seen in Table 3, the best-performing condition was highly variable. The overall BPref for the maximum from a given condition was 0.4910.

Table 1 – The official topic description, the manual query (Boolean operators in CAPS, with AND having precedence and phrases in parentheses), and the ICD-9 query for all topics.

Topic	Description	Textual Query	ICD-9 Query
101	Patients with hearing loss	(hearing loss) OR deaf	389.*
102	Patients with complicated GERD who receive endoscopy	GERD OR (gastroesophogeal reflux) AND (endoscopy or EGD)	530.11
103	Hospitalized patients treated for methicillin-resistant Staphylococcus aureus (MRSA) endocarditis	(methicillin-resistant Staphylococcus aureus) OR (MRSA) AND endocarditis	041.1*
104	Patients diagnosed with localized prostate cancer and treated with robotic surgery	(prostate cancer) AND (robotic surgery)	185
105	Patients with dementia	dementia OR alzheimer's	290.*
106	Patients who had positron emission tomography (PET), magnetic resonance imaging (MRI), or computed tomography (CT) for staging or monitoring of cancer	(positron emission tomography) OR PET OR (magnetic resonance imaging) OR MRI OR (computed tomography) OR CT AND (staging OR monitoring) AND cancer	(no code)
107	Patients with ductal carcinoma in situ (DCIS)	(ductal carcinoma in situ) OR DCIS	233.0
108	Patients treated for vascular claudication surgically	claudication AND surgery OR surgical	440.21
109	Women with osteopenia	osteopenia OR bone loss	733.*
110	Patients being discharged from the hospital on hemodialysis	discharge AND hemodialysis	39.95
111	Patients with chronic back pain who receive an intraspinal pain-medicine pump	(back pain) AND intraspinal AND pump	724.*
112	Female patients with breast cancer with mastecomies during admission	(breast cancer) AND mastectomy	174.*
113	Adult patients who received colonscopies during admission which revealed adenocarcinoma	colonoscopy AND adenocarcinoma	153.*
114	Adult patients discharged home with palliative care / home hospice	discharge AND (palliative care) OR hospice	V66.*
115	Adult patients who are admitted with an asthma exacerbation	asthma AND exacerbation AND admi*	493.0* or 493.1*
116	Patients who received methotrexate for cancer treatment while in the hospital	methotrexate AND cancer	(no code)
117	Patients with Post-traumatic Stress Disorder	(post-traumatic stress disorder) or PTSD	309.81
118	Adults who are received a coronary stent during an admission	coronary AND stent	414.0*
119	Adult patients who presented to the emergency room with with anion gap acidosis secondary to insulin dependent diabetes	(anion gap) AND acidosis AND (diabetes mellitus) or IDDM	250.1*
120	Patients admitted for treatment of CHF exacerbation	(congestive heart failure) OR CHF AND admi*	428.0
121	Patients with CAD who presented to the Emergency Department with Acute Coronary Syndrome and were given Plavix	(acute coronary syndrome) OR (coronary artery disease) OR CAD AND plavix	414.0*

122	Patients who received total parenteral nutrition while in the hospital	(total parenteral nutrition) OR TPN	99.15
123	Diabetic patients who received diabetic education in the hospital	(diabetes mellitus) OR diabetic OR DM AND education	250.*
124	Patients who present to the hospital with episodes of acute loss of vision secondary to glaucoma	(acute vision loss) AND glaucoma	365.*
125	Patients co-infected with Hepatitis C and HIV	(hepatitis C) OR HCV AND (human immunodeficiency virus) OR HIV	(070.4* or 070.5*) AND (042 or 043)
126	Patients admitted with a diagnosis of multiple sclerosis	(multiple sclerosis)	340
127	Patients admitted with morbid obesity and secondary diseases of diabetes and or hypertension	(diabetes mellitus) OR diabetic OR DM OR hypertension AND (morbid obesity)	278.01 and (250.* or 401.* or 405.*)
128	Patients admitted for hip or knee surgery who were treated with anti-coagulant medications post-op	(hip surgery) OR (hip replacement) OR (knee surgery) OR (knee replacement) AND anti-coagulant OR heparin OR warfarin OR coumadin	(iffy) 99.19 and (81.5* or 80.*)
129	Patients admitted with chest pain and assessed with CT angiography	(chest pain) AND (ct angiography)	786.5*
130	Children admitted with cerebral palsy who received physical therapy	(cerebral palsy) OR CP AND (physical therapy) OR PT	343.*
131	Patients who underwent minimally invasive abdominal surgery	(minimally invasive) AND (abdominal surgery)	(no code)
132	Patients admitted for surgery of the cervical spine for fusion or discectomy	(cervical spine) AND fusion OR discectory	(no code)
133	Patients admitted for care who take herbal products for osteoarthritis	osteoarthritis AND herbal	715.*
134	Patients admitted with chronic seizure disorder to control seizure activity	(seizure disorder) AND control	345.*
135	Cancer patients with liver metastasis treated in the hospital who underwent a procedure	liver AND metastasis AND procedure	155.2 or 197.7

Table 2 – Results of Bpref and Precision @ 10 visits for OHSU runs. The first two rows are the officially submitted runs, while the following six rows represent our post-conference runs.

Run	Bpref	Precision @ 10
ohsuManAll	0.3792	0.582
ohsuManLim	0.3060	0.589
Text-only – All	0.3751	0.5853
Text-only – Lim	0.2894	0.4824
Text AND ICD-9 – All	0.2497	0.4471
Text AND ICD-9 – Lim	0.1695	0.3235
Text OR ICD-9 - All	0.3657	0.4618
Text OR ICD-9 – Lim	0.3238	0.4206

Table 3 – Run that achieved best Bpref results by topic. When more than one run achieved the best results, the amount (1) was divided among them. The results show that the Text OR ICD-9 – Lim run had the best results most often, but that each run had at least a few instances of the best Bpref on a given topic.

Topic	Text-Lim	Text-All	AND-Lim	AND-All	OR-Lim	OR-All
101					1	
102						1
103	1					
104				1		
105	1					
106		0.5				0.5
107						1
108		0.5				0.5
109		1				
110	0.5				0.5	
111	1					
112						1
113						1
114	1					
115						1
116		0.33		0.33		0.33
117						1
118			1			
119						1
120			1			
121		1				
122	0.5				0.5	
123	1					
124						1
125						1
126			0.25	0.25	0.25	0.25
127					1	
128		0.5				0.5
129		1				
131		0.33		0.33		0.33
132		0.33		0.33		0.33
133						1
134						1
135					1	
all	6	5.49	2.25	2.24	4.25	13.74

The official NIST evaluation for the track focused on two metrics: Bpref and early precision (P@10). For our official runs, the BPref of ohsuManAll was slightly below the median at 0.3792 (vs. 0.412), while ohsuManLim was even worse at 0.306. For P@10, however, both of our runs outperformed the average median of 0.476; ohsuManAll had a P@10 of 0.582 while ohsuManLim had a P@10 of 0.509. Nonetheless, given the task description, P@10 is not really the most appropriate measure of comparison. Identifying sets of patients for a research study is a task for which recall is more important than precision. The Bpref measure more accurately reflects performance over a wider range of retrieval.

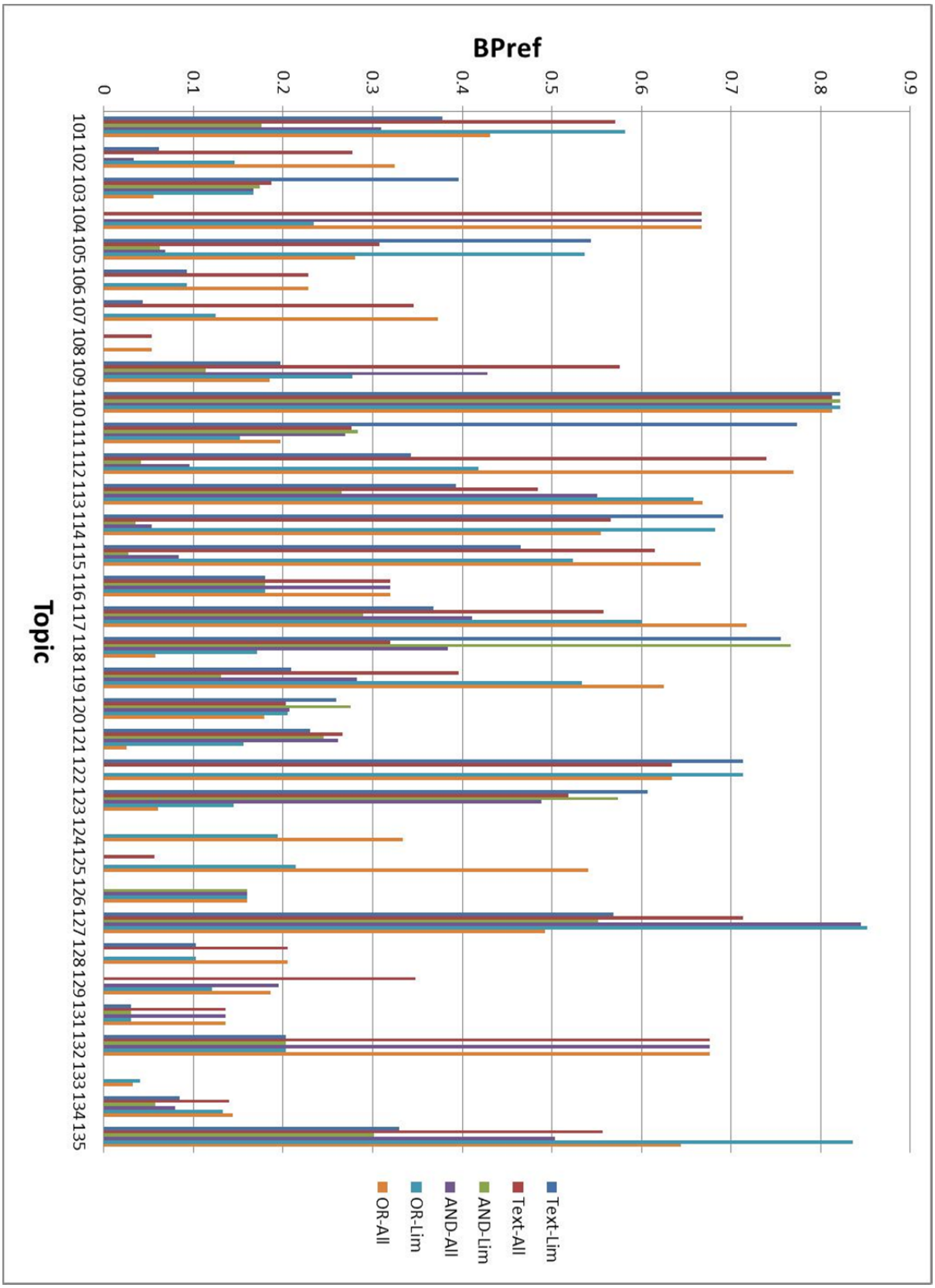

Figure 2 – Results of each run per topic.

Given the wide variety of topics in this year's challenge—both in terms of lexical as well as semantic complexity—depending on such aggregated statistics to evaluate a system's performance can obscure differences among topics. Indeed, our system's performance varied widely across topics and also between runs, as shown in Figure 2. In terms of Bpref, the best topic for both of our runs was #110 ("Patients being discharged from the hospital on hemodialysis"), for which our system achieved a Bpref of 0.8132 and 0.8213 for ohsuManAll and ohsuManLim, respectively.

The topic for which our system had its worst performance was topic #124 ("Patients who present to the hospital with episodes of acute loss of vision secondary to glaucoma"). Neither run returned any relevant documents. In addition to topic #124, each run encountered its own set of challenging topics. ohsuManAll only included one document for topic #133 ("Patients admitted for care who take herbal products for osteoarthritis"), which did not happen to be relevant; ohsuManLim failed to retrieve any relevant articles at all for topics #125 and #129 ("Patients co-infected with Hepatitis C and HIV" and "Patients admitted with chest pain and assessed with CT angiography" respectively). Interestingly, while these two topics were among the worst for ohsuManLim, ohsuManAll performed at median for #125 and at nearly double the median for #129.

Looking at our post-conference runs, we found that using a Boolean AND to combine textual queries and ICD-9 codes nearly always (in 30 out of 35 topics) resulted in decreased Bpref. The five runs that did see an increase in Bpref almost all saw just a small increase. Using OR with text queries and ICD-9 codes generally hurt Bpref, but not to the same degree using AND. In this instance, 13 of the 35 topics actually exhibited an increased Bpref. In some cases, the OR of ICD-9 codes with the text queries resulted in substantial gains in Bpref. The OR of the ICD-9 wildcard with the topic #113 textual query, for example, resulted in an increase in Bpref from 0.49 to 0.67, along with an increase in the number of retrieved visits from 55 to 79.

4. Discussion

Our two official runs performed quite differently. In terms of Bpref, ohsuManAll was generally better, and outperformed ohsuManLim in 21 of the 34 scored topics. Often this difference in performance was substantial, as there were eight topics for which the Bpref of ohsuManAll was double that of ohsuManLim. There was one topic (#107, "Patients with ductal carcinoma in situ (DCIS)") for which the Bpref of ohsuManAll Bpref was much larger than that of ohsuManLim.

There were, however, several topics for which ohsuManLim outperformed ohsuManAll by a notable degree. For example, in topics #118 ("Adults who received a coronary stent during an admission") and #103 ("Hospitalized patients treated for methicillin-resistant Staphylococcus aureus (MRSA) endocarditis"), the Bpref of ohsuManLim was nearly double that of ohsuManAll.

In many cases, there were reasonable explanations for a topic's divergent performance. For example, in the case of topic #107, the dramatic seven-fold difference in Bpref between the two runs was due to the fact that ohsuManLim included only two visits, whereas ohsuManAll included 10, eight of which were relevant. The smaller number of visits included in ohsuManLim was most likely due to the fact that the search of ohsuManLim specifically excluded surgical pathology reports, which was where many potentially relevant passages of text would have occurred. Of course, the appropriate document type

sources of information may vary greatly by topic even in a system focused on identifying appropriate patients such as this.

In addition to comparing our two runs to one another, we also compared our runs' performance to that of the rest of the participants. As mentioned previously, our runs generally underperformed relative to the median in terms of Bpref, but performed better in terms of P@10. When we looked at this on a topic-by-topic basis, we saw that there were several topics for which our system strongly outperformed the median; interestingly, the two topics for which we included explicit ICD-9 code criteria into the search query were among those that had the greatest difference in Bpref performance (topics #111 and #117).

However, there were several topics for which our system's performance was low compared with the median. For example, our system did not do very well on topics #105 ("Patients with dementia") or #120 ("Patients admitted for treatment of CHF exacerbation"). Our runs for both of these topics included quite a large number of nonrelevant documents and, while they both had quite high early precision, their later precision was extremely low. Since the relevant documents are correctly ranked higher than the retrieved non-relevant documents, our system may benefit from future work in which an appropriate cut-off value is estimated and applied.

Looking at our follow-on runs, we found that using a Boolean AND to combine textual queries and ICD-9 codes nearly always (in 30 out of 35 topics) resulted in decreased Bpref, with the five runs that did see an increase in Bpref almost all seeing a very small increase. The OR of textual queries with the ICD-9 codes generally hurt Bpref, but not to the same degree as their combination with AND. In some cases, the OR of textual queries with ICD-9 codes resulted in substantial gains in Bpref. For example, an OR with the ICD-9 wildcard query for topic 113 resulted in an increase from 0.49 to 0.67 (along with an increase in the number of retrieved visits from 55 to 79).

In general, the benefit gained by including ICD codes in a topic's query seems to depend heavily on the specific nature of the topic. In future work, it may be possible to classify topics automatically as to whether or not including ICD codes would be of value. To help address this in the future, we may explore automated query expansion techniques, as well as the use of techniques to identify similar documents given a small number of "seed" documents. Integrating these sorts of support tools into our interactive search system could help our users handle situations in which keyword-based searching was insufficient.

5. References

1. Safran C, Bloomrosen M, Hammond WE, Labkoff SE, Markel-Fox S, Tang P, et al., *Toward a national framework for the secondary use of health data: an American Medical Informatics Association white paper.* Journal of the American Medical Informatics Association, 2007. 14: 1-9.
2. Friedman CP, Wong AK, and Blumenthal D, *Achieving a nationwide learning health system.* Science Translational Medicine, 2010. 2(57): 57cm29.
3. Blumenthal D, *Launching HITECH.* New England Journal of Medicine, 2010. 362: 382-385.