Cohort Shepherd: Discovering Cohort Traits from Hospital Visits

Travis Goodwin, Bryan Rink, Kirk Roberts, Sanda M. Harabagiu
Human Language Technology Research Institute
University of Texas at Dallas
Richardson TX, 75080
{travis,bryan,kirk,sanda}@hlt.utdallas.edu

Abstract

This paper describes the system created by the University of Texas at Dallas for content-based medical record retrieval submitted to the TREC 2011 Medical Records Track. Our system builds a query by extracting keywords from a given topic using a Wikipedia-based approach we use regular expressions to extract age, gender, and negation requirements. Each query is then expanded by relying on UMLS, SNOMED, Wikipedia, and PubMed Co-occurrence data for retrieval. Four runs were submitted: two based on Lucene with varying scoring methods, and two based on a hybrid approach with varying negation detection techniques. Our highest scoring submission achieved a MAP score of 40.8.

1 Introduction

The 2011 Text REtrieval Conference (TREC) Medical Records Track evaluates the effectiveness of providing content-based retrieval for electronic medical records (EMRs). Participants were given a set of EMRs from the University of Pittsburgh BLULab NLP Repository[1] as well as a mapping between hospital visits and medical records. Additionally, we have been provided with a set of sample topics. These topics are based on a list of "priority areas" created by the Institute of Medicine (CCERP and Institute of Medicine). Each topic targets certain cohorts – groups of people sharing a common attribute – and is designed to find a population over which comparative effectiveness studies can be done. Examples of topics used for training are provided in Table 1.

1. *Patients taking* ***atypical antipsychotics*** *without a diagnosis* ***schizophrenia*** *or* ***bipolar depression***
2. *Patients treated for* ***lower extremity chronic wound***
3. *Patients with* ***atrial fibrillation*** *treated with* ***ablation***
4. *Elderly patients with* ***ventilator-associated pneumonia***

Table 1: Provided sample topics with keywords highlighted in bold.

The goal of this track is to return a ranked list of hospital visits that satisfy the requirements expressed in each topic. A hospital visit is a set of electronic medical records that pertain to a single patient's visit to the hospital. As each hospital visit contains multiple EMRs (as many as 415), producing a ranked list of hospital visits is much more complicated than retrieving a ranked list of individual documents when using a query as complex as a topic. Moreover, hospital visits may consist of multiple types of EMRs, e.g. an operating room report, multiple radiology reports, a discharge summary, and other reports detailing physical findings, plans of treatments, descriptions of the patient's problem, or laboratory test results. In a hospital visit, many EMRs are generated for a patient, but only a few of them may be relevant to the topic of interest. Because of this, the content-based retrieval system that we built for this evaluation operates at the hospital visit level instead of the EMR level.

Given the wealth of medical information that is currently available, we incorporate medical information from the Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) and the Unified Medical Language System (UMLS). We also rely on the knowledge encoded in PubMed Central and Wikipedia. These forms of knowledge greatly contribute to our system.

The remainder of this paper is outlined as follows. Section 2 provides an overview of our approach to

[1] This collection is available at the following URL: *http://www.dbmi.pitt.edu/nlpfront.*

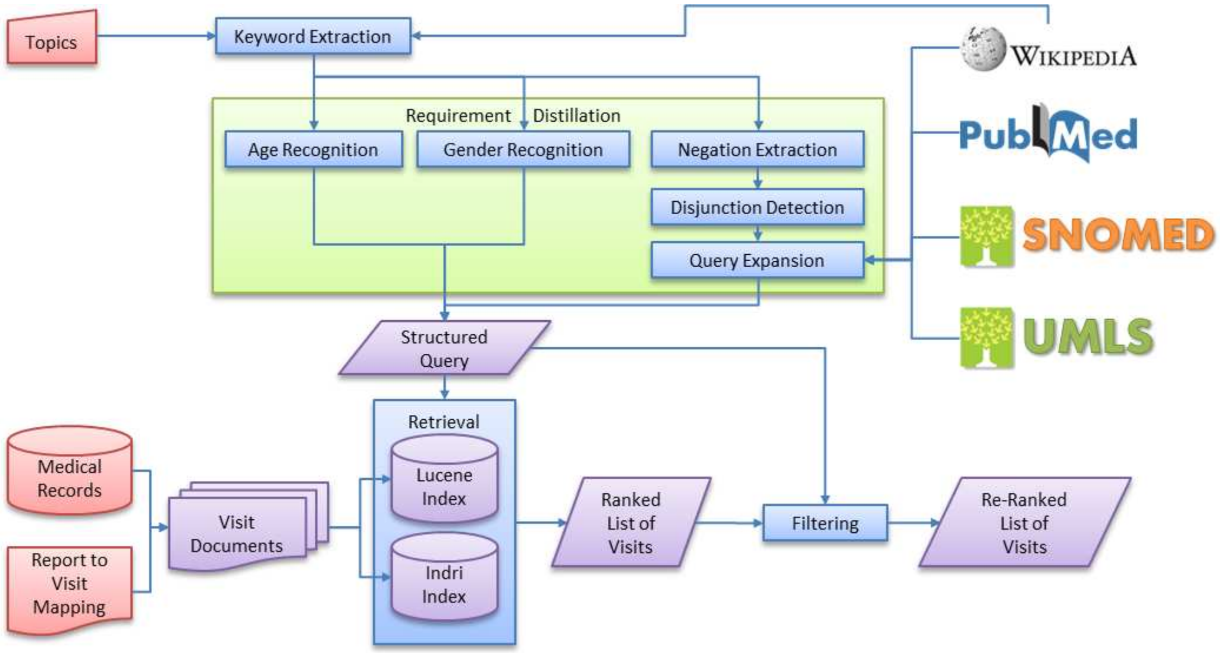

Figure 1: The Architecture of Cohort Shepherd

this task and Section 3 explains how we extract keywords from a given topic. Section 4 expresses our methods for extracting additional requirements from the topic, with sub-sections detailing each type of requirement. Section 5 provides a discussion on our methods for expanding the keywords we extracted while Section 6 illustrates our three methods of retrieving and ranking relevant hospital visits. Section 7 outlines how we filter our initial list of hospital visits based on our extracted topic requirements. Finally, Section 8 provides an evaluation of the performance of each of our four submissions, Section 9 offers a discussion on ours results, and we conclude our paper with Section 10 by reviewing the task and our approach, and by looking to the future.

2 The Architecture

Our approach converts each topic into a machine-readable query. These queries are generated by extracting all keywords that capture the essential requirements of each topic. The keywords extracted from the provided sample topics are given in bold in Table 1. Many of the topics expressed additional requirements such as age restrictions (e.g. *elderly patients*, *patients younger than 30*), gender restrictions (e.g. *male patients*, *women*), keyword negations (e.g. *not diagnosed with schizophrenia*) or disjunctive keywords (e.g. *diagnosed with schizophrenia or bipolar depression*). After keywords have been extracted, any of these additional requirements present in the topic are detected and incorporated within the topic's associated query.

The medical records used for this task vary significantly in terminology when discussing any given keyword. For example, the keyword *atypical antipsychotics* appears in only three medical records. Thus, it is not enough to merely find documents that contain the detected keywords themselves; rather, we must expand each keyword to include other words that denote the same idea: synonyms, hyponyms, and alternate phrasings and spellings. To this end, we expand each keyword using information contained in the UMLS (Bodenreider, 2004) and SNOMED (Stearns et al., 2001) medical terminology databases, as well as information from Wikipedia and the PubMed Central Open Access Subset.

Unlike query formation, our approach to retrieval, scoring, and filtering varies slightly between our four submissions. Our first two submissions use the Apache Lucene (Hatcher and Gospodnetic, 2005) information retrieval system: we created a Lucene index for the provided medical records, and each query is converted into a Lucene-understandable format for retrieval. Our last two submissions also use the Indri (Strohman et al., 2004) language model-based search engine in conjunction with Lucene. To resolve the problem of retrieving visits (sets of medical records corresponding to a single patient's hospital

Keyword Requirements
A required keyword must:

1. Occur less than 30,000[3]times in the PubMed Central Open Access Subset[4]collection.
2. Not start with one of eleven invalid start words (provided in Table 2).
3. Not contain one of thirty-two medical stop words (provided in Table 3).
4. Be longer than one character in length, and contain at least one non-punctuation character.

Figure 2: Conditions applied to extracted keywords.

visit) our system merges all the records associated with each visit into a single document when indexing. Our queries are processed and ranked by the given retrieval system(s). We then re-rank and filter these preliminary results to ensure that any additional, non-keyword requirements – age, gender, and negation – are taken into account.

a	any	her	the	some	to
their	every	this	all	his	

Table 2: Words not allowed to begin any keyword

visit	session
visits	sessions
report	rendezvous
reports	rendezvouses
patient	record
patients	records
doctor	hospital
doctors	hospitals
nurse	hospitalization
nurses	hospitalizations
encounter	stay
encounters	stays
appointment	note
appointments	notes
meeting	course
meetings	courses

Table 3: Medical stop words.

3 Keyword Extraction

Because each topic targets a specific cohort, a group of people satisfying certain conditions, we must convert these conditions into a machine-readable format. To do this, we first extract all keywords – sequences of terms indicating the major concepts or conditions – from the topic. Using the OpenNLP [2] phrase chunker, each extracted noun phrase is scanned for keywords.

The longest sequences of words within each noun phrase that correspond to the title of an existing Wikipedia article and match the conditions given in Figure 2 are taken as keywords. For example, the noun phrase NP = *lower extremity chronic wound* does not correspond to an existing Wikipedia article. However, the subsequences of that phrase, s_1 = *lower extremity* and s_2 = *chronic wound* are both titles of Wikipedia articles. Thus, a topic containing the noun phrase NP will have the keywords *lower extremity* and *chronic wound* extracted.

Because the presence of a noun phrase in a topic nearly always indicates a condition of the cohort in question, we use a secondary method for detecting keywords if our Wikipedia-based approach found none. Because in English the head of a noun-phrase is typically the right-most word, we take the longest sub-sequence ending in the right-most word that satisfies the conditions in Figure 2 as a keyword.

Any keyword detected in either of these two methods is assumed to represent a requirement upon the cohort targeted in the topic. Any visit considered relevant must contain occurrences of all required keywords. Unfortunately, not all conditions are expressed in noun phrases, and not all noun phrases are properly detected. To address this, we examine each word in the topic which was not used as part of a keyword to determine if it should be used as an optional, or non-required keyword. Any word that satisfies the keyword conditions in Figure 2 and the additional, stricter requirement that the word may not occur more than 10,000 times in the PubMed Central Open Access Subset is taken as a non-required keyword. These optional keywords capture additional constraints not detected in noun phrases of the topics, but are too noisy to be considered as requirements. Thus, optional keywords are used merely as indicators of potential relevance: visits that contain mentions of an optional keyword are more likely to be more relevant than those that do not, however the absence of an optional keyword does not prevent a visit from being relevant. For example, in the topic *patients admitted with new-onset* **diabetes**, diabetes is detected a required keyword, but *new-onset* is detected an optional keyword.

[2] OpenNLP is an open source natural language processing tool suite, available at *http://incubator.apache.org/opennlp/*. We used the provided English model trained on conll2000 shared task data.

[3] A threshold based on inspection of occurrences of common terms in the PubMed Central Open Access Subset.

[4] Described further in section 5.4.

4 Distillation of Topic Requirements

Although many of the requirements expressed in the training topics were captured by the keywords we extracted, some topics required additional requirements, such as age restrictions (e.g. ***teenagers*** *who have taken or plan to take Plan B*) and gender restrictions (e.g. ***men*** *with prostate cancer treated with surgery or radiotherapy*. Consequently, we devised a requirement distillation method that captures these additional and important characteristics of the topics.

Additionally, topics included disjunctions or negations that are ignored by the keyword extraction technique that we developed. To address this limitation, we have developed four distinct methods for recognizing topic requirements.

4.1 Extraction of Patient Age Requirements

Topics such as ***elderly*** *patients with ventilator-associated pneumonia* or *patients* ***in their 20s and 30s*** *admitted for overdose* pose an additional requirement: hospital visits returned should focus on patients whose age lies within a certain range. These age restrictions are extracted with the following grammar:

```
<age-phrase>
  ::= <unqualified-prefix><num><age-qualifier>
     | <qualified-prefix><num>
     | <num><age-qualifier><unqualified-suffix>
     | <num><qualified-suffix>
     | <range-prefix><num><range-infix><num>
     | <known-age-entity>
```

Where a <num> entity captures both English and numeric representations of ages, and a <known-age-entity> is one of a few dozen manually created classes with known age ranges, such as *elderly* (ages 60 or older), *children* (aged 2 to 12), or *adult* (aged 20 or older).

To ensure the captured <num> describes an age, and not a range of some other, arbitrary domain, each rule requires either a qualified prefix or suffix, or an age qualifier. For example, the sequence *patients younger than 30* contains the qualified prefix *younger than* denoting that the captured number, 30, is a description of age. The sequence *patients of at most 30*, by contrast, must be followed by an age qualifier such as *years* to establish that the captured range denotes an age range, and not, say, a BMI (body mass index) range. If an age range is found, the age requirements are stored as part of the query. Additionally, if any keyword extracted for this topic conveys the same patient requirements as the extracted age range (e.g. the keyword *children*), the keyword is discarded because the age requirement already conveys this restriction more directly.

4.2 Extraction of Patient Gender Requirements

In addition to patient age requirements, some topics expressed gender requirements upon targeted patients. For example, the topic *men with prostate cancer treated with surgery* imposes the requirement that all returned hospital visits pertain to male patients. We extract such genders requirements using simple regular expressions that search for an occurrence of one of 17 gender-indicating words created for both genders (e.g. words that indicate the male gender are *man*, *men*, *boy*, *he*, etc.). If a gender word is detected in a topic, the gender requirement is stored in the query. If more than one gender was detected in a topic (i.e. if both the words *women* or *men* occur), the associated query is assumed to have no gender requirements. As with age requirement extraction, keywords that indicate redundant patient traits to the extracted gender are removed. Thus, the possible gender requirements are 'male', 'female', or 'either'.

4.3 Detection of Negations

Many topics use negation; thus, the scope of negation needs to be detected so that keywords that were extracted from within a negation scope can be negated as well. We check for any negations upon our extracted keywords. For example, the topic *patients with cancer* ***not treated with surgery***, requires that all corresponding patients were not treated with surgery. Such negations within the topic are detected, for the first three runs, using the NegEx regular expression system which is based on regular expressions (Chapman et al., 2001). The fourth run uses the LingScope system and is described in Section 6.2.2 of this paper. Both systems output negation "scopes" which are word sequences from the topic that contain negated terms. Each topic is processed for negations by comparing each extracted keyword to each negation scope. Any keyword found within a negation scope is marked as negated keyword in the query.

4.4 Detection of Disjunctions

Because the topics are expressed in natural language, syntactic complexities arise. One such complexity is encompassed by co-ordinated disjunctions, After processing for keyword negations, we check for disjunctions within the topic, such as *surgery or radiotherapy.* We detect these keyword disjunctions using the Stanford Dependency Parser (De Marneffe et al., 2006). This is accomplished by analyzing the dependency parse for any conjunction dependencies with the word "or", denoted as *conj_or* in the annotations. If a disjunction is found, any keywords corresponding to either disjunctive dependency is marked as a disjunctive keyword. Each disjunctive keyword is stored along with its disjunctive set: the set of other keywords that form a disjunction with the keyword. For example, given the topic *men with prostate cancer treated with* ***surgery or radiotherapy***, both *surgery* and *radiotherapy* would be marked as disjunctive keywords, and each would record the disjunctive set {*surgery*, *radiotherapy*}.

5 Query Expansion

In written text, and especially in medical records, the morphology of words varies significantly both between and within documents. In order to account for these variations in text, we store each extracted keyword in several forms. All queries internally store up to six variations of each keyword: its originally detected form, a WordNet (Fellbaum, 1998) lemmatized form, an unabbreviated form (based on a list of common medical abbreviations), a form in which hyphens are padded with spaces, a form with all hyphens replaced with spaces, and a form with all punctuation removed. These forms are used as synonyms for the purposes of keyword expansion, negation filtering, and retrieval.

These slight variations in morphology are not enough to capture the diverse ways in which keywords may be expressed in medical records. Indeed, the topics presented in this task often require extensive domain knowledge within the field of medicine in order for some keywords to be properly understood. For example, a keyword such as *hearing loss* may be referred to as *hearing impairment*, *hard of hearing*, *decreased hearing*, or *difficulty hearing.* These synonyms all denote hearing loss, but use alternate phrasings. Unfortunately, some keywords are often represented with more than synonyms. For example, the keyword *atypical antipsychotics* almost never occurs in any of the medical records. Instead, hyponyms (more specific words) such as *zoloft*, *seroquel*, or *abilify* are used instead. In our approach, we consider all of these semantically related words as 'expansions', and we refer to the process of generating them as keyword or query expansion. In addition generating these expansions, we also assign weights, or confidences, for each expansion based on intuition regarding the nature of the resource used. Future work may find more value in automatically setting (or learning) these weights, but due to limitations in both time and training data, our weights were decidedly manually.

After each expansion phase, keywords are analyzed to ensure that each expanded term is only retained for at most one keyword. For example, if, both *atrial fibrillation* and *ablation* contain *heart* as a synonym, it would be retained as an expansion only for the keyword which has it with the highest weight. This ensures that no keyword may eclipse another keyword with its expansions. Additionally, any keyword which expands into another keyword's original form (unexpanded form) is merged with that keyword, because both keywords are assumed to be direct synonyms. For example, the topic *women who are currently pregnant and have been smoking and/or drinking during the pregnancy*, contains the keywords *pregnant* and *pregnancy*, which both contain each other in their expanded terms. Thus, both *pregnancy* and *pregnant* would be merged into a single keyword that contains the union of each keyword's expanded terms and forms.

We expand each keyword with using four techniques, which are explained in the following subsections.

5.1 UMLS Expansion

The Unified Medical Language System (UMLS) is a resource for coordinating health and medical vocabularies. UMLS contains three major components: the "Metathesaurus" (which includes data from SNOMED, RxNorm, MeSH, and other collections), the "Semantic Network" which provides general categories and relationships, and the "SPECIALIST Lexicon and Lexical Tools"[5]. Our system uses the UMLS Metathesaurus to generate high confidence synonyms: each keyword is expanded to include all concepts in the Metathesaurus which share the same UMLS concept ID as the keyword (an abridged example is provided in Table 4). Each expansion added by UMLS expansion is assigned a weight of 12.

5.2 Wikipedia Redirect Expansion

Wikipedia is a common resource for natural language processing tasks. The English version is comprised of 3,731,340 user-generated articles covering almost any notable topic[6]. In addition to articles, Wikipedia also contains pages called *redirects* which do not con-

[5] UMLS is described at *http://www.nlm.nih.gov/research/umls/quickstart.html.*

[6] The number of articles is based on September 6, 2011.

UMLS Expansions	
Original	Stroke
Expansions	apoplexy brain attack vascular accident, brain cerebrovascular accident accident - cerebrovascular
Original	Lower Extremity
Expansions	hindlimb hind limb lower limb leg leg region

Table 4: An example of select UMLS expansions.

tain content themselves, but rather redirect – send – the reader to another article (or section of an article). Redirects typically embody alternate names, spellings, forms, closely related words, alternately punctuated or encoded forms, less specific forms in which the redirected name is the primary topic, or more specific forms of some other page. Our system uses Wikipedia redirects[7] to generate synonyms, and alternate (or mis-) spellings for keywords. We do this by expanding each keyword so that it includes all article titles that redirect (send the user) to the same article as the keyword. Any expansions added this way are assigned a weight of 10.

Wikipedia Expansions	
Original	Hearing Loss
Expansions	auditory impairment deaf deafness hard of hearing hearing damage
Original	Ablation
Expansions	ablate ablated ablative cooling ablative material rotoablation

Table 5: An example of select Wikipedia expansions.

5.3 SNOMED CT Expansion

The National Library of Medicine provides a resource called the Systematized Nomenclature of Medicine–Clinical Terms (SNOMED CT). SNOMED CT is a clinical terminology being maintained by the International Health Terminology Standards Development Organisation and is one of several designated standards used in United States Federal Government systems for the electronic exchange of medical records[8]. SNOMED CT catalogs both medical concepts and various relationships between them. SNOMED CT is incorporated into our system so that we may generate hyponym (more specific term) expansions for each keyword. To generate these hyponyms, we include all SNOMED CT concepts that partake in the child side of an *is_a*, *part_of*, or *component* relationship[9] as expansions of a keyword. All concepts added by SNOMED CT expansion are given a weight of 8.

SNOMED CT Expansions	
Original	Atypical Antipsychotics
Expansions	abilify aripiprazole asenapine clozapine clozaril
Original	Dementia
Expansions	alzheimer's disease vascular dementia dialysis dementia neurosyphilis postconcussion syndrome

Table 6: An example of select SNOMED expansions.

5.4 Co-occurrence Expansion

Because all keywords may not have clear synonyms or hyponyms, our final method of expansion finds related terms for each keyword. We use these related words as a fall-back so that if a keyword lacks helpful expansions from the previous techniques, we may at least find documents that share similar context with our keywords.

The keyword *lower extremity*, for example, does not have any hyponyms in UMLS. Fortunately, our co-occurrence expansion approach generates many words that, while not direct synonyms or hyponyms, strongly imply the context of a lower extremity. For *lower extremity*, our approach generates expansions including *ankle*, *amputation*, *tibial* (related to the shinbone), *popliteal* (related to the hollow at the back of the knee), *thigh*, and *toe*. We generate these expansions by calculating the semantic similarity between n-grams (up to size 2) and the keyword and picking the most similar n-grams. We measure this

[7] Redirect and article data is based on the May 26, 2011 English Wikipedia data dump.

[8] More information on SNOMED CT is available at *http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.*

[9] We consider children up to levels (grandchildren) from the parent concept.

semantic similarity using the normalized Google distance(Cilibrasi and Vitanyi, 2007):

$$NGD(k,t) = \frac{\max\left(\log o(k), \log o(t)\right) - \log c(k,t)}{\log N - \min\left(\log o(k), \log o(t)\right)}$$

Where k represents the given keyword, t represents an n-gram, and $o(x)$ is a function providing the number of documents containing term x, $c(k,t)$ is a function yielding the number of documents containing both keyword k and term t, and N is the number of documents in the collection. An NGD of zero denotes terms that always appear together while infinity denotes terms that never appear together (Cilibrasi and Vitanyi, 2007).

While NGD was originally created to operate on hit counts returned from the Google search engine, our occurrence (and co-occurrence) information is based on the PubMed Central Open Access Subset, a small portion of articles in PubMed Central available under a Creative Commons license, henceforth simply refereed to as PMC[10]. PMC consists of full-text biomedical and life sciences journal literature from the United States National Institutes of Health's National Library of Medicine. The version of this corpus we used contains 234,591 documents. We calculate the occurrences for each term as the number of PMC documents that contain the term; likewise, we calculate the co-occurrences of terms x and y as the number of documents that contain both terms x and y. For each keyword, k, we rank every n-gram (term), t, seen in the corpus based on the score $NGD(k,t)$. From that ranking, the most similar m_k terms are retained as expansions of keyword k.

Because NGD provides a distance between all words that occur in the same document as any keyword, we must create an upper bound on the number of these words we used. We do this by sorting the list of related n-grams, and bounding that list by the equation

$$m_k = \log \frac{o(k)}{20}$$

which limits the number of expansions of each keyword by the frequency of the keyword in PMC. This formula was chosen after inspecting ranked NGD expansions for several keywords from the training topics. m_k expresses our observation that more common keywords often had a greater number of useful expansions.

Finally, we give a weight, w_k to each expansion based on its NGD similarity:

$$w_k(t) = \left(\frac{1.0 - NGD(k,t)}{z}\right)^{\eta}$$

Where z is the observed upper bound for NGD similarity and η exaggerates the differences between high and low NGD scores (we set $\eta = 5$). This adjustment accounts for our experience that that the usefulness of these expansions diminishes quickly when $NGD > 0.30$ and that terms within a distance of 0.075 are typically inflected forms of the keyword and thus highly related. w_k emphasizes the importance of highly related terms while diminishing the weight of terms with have a further distance. Table 7 contains examples for the first five expansions of the keyword *atypical antipsychotics*, which are all types of antipsychotic medications.

Co-occurrence Expansions				
Atypical Antipsychotics				
Expansion	$o(k)$	$o(t)$	$c(k,t)$	w
olanzapine	397	520	236	.76
risperidone	397	527	222	.71
quetiapine	397	347	164	.69
clozapine	397	440	177	.68
antipsychotic drug	397	210	82	.64

Table 7: An example of NGD expansions and their associated weights.

6 Relevant Visit Retrieval

6.1 Lucene-based Approaches

Our first two submissions both use Apache Lucene (Hatcher and Gospodnetic, 2005) as their information retrieval system. Reports are indexed at a visit-level – that is, all records are merged into a single document for each hospital visit. Retrieval is done by converting each keyword into a Lucene Boolean query of keyword forms and expansions, boosted according to their weights. In both approaches, all Lucene scores are normalized by setting each score to the ratio of that score to the highest score for the query so that the top-ranking visit has a score of 1.0.

6.1.1 Standard Lucene Approach

Our first run (*UTDHLTSL*) uses standard Lucene scoring, which calculates the cosine-distance between each visit and the given query vector. In this run, each question is converted into a Lucene query containing each required keyword's sub-query as a required clause. In this way, a document's score is equal to the sum of its scores for each keyword's sub-query.

[10] The PMC Open Access Subset is available at *http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/.*

This has the effect of allowing documents which have a high score for one keyword and a very low score for another to outweigh documents that have a moderate score for both.

6.1.2 Multiple Keyword Focused Lucene Approach

Our second run (*UTDHLTMK*) attempts to mitigate the ability for a single keyword to eclipse the scores for other required keywords by splitting each query into separate queries for each keyword, and combining the scores for each of these sub-queries in the following way:

$$\begin{aligned} score(q,d) = &\sum_{k \in R} Lucene(query(k), d) \\ &\times \prod_{k \in R} Lucene(query(k), d) \\ &+ |R| \end{aligned}$$

Where $score(q, d)$ is the score for document d relative to query q, R is the set of keywords found in both q and d, $Lucene(x, d)$ is Lucene's score for query x and document d, and $query(k)$ is the Lucene query for keyword k. This has the effect of partitioning the results into tiers based how many keywords were matched in each document, represented by the significand, while the fractional portion represents the strength of all occurring keywords within that document. The product of sub-queries provides a stronger indicator of the relevance for all keywords than the sum, as the score depends more closely on each keyword's score. However, because Lucene is indexing visits which are actually sets of (often diverse) documents, document length is rarely reflective of a visit's relevance to a given query. As a result, visits containing very few reports tend to have much higher scores. To that end, the sum and product of the scores for each sub-query are combined to lessen the significant effect of a document's length on its relevance.

6.2 Hybrid Approaches

Our last two runs use a combination of Indri (Strohman et al., 2004) and the two previously mentioned Lucene approaches to score documents. Scoring is done by a weighted vote between each system, with the weights provided in Table 8. These weights are based on our intuition and scaled to normalize scores between systems. When processed by Indri, each query is represented as a weighted query combining the scores for all keywords, where each keyword is represented as a weighted synonym list.

Weights for Combined IR Systems	
System	**Weight**
Standard Lucene	0.6
Keyword Focused Lucene	$\lvert keywords \rvert^{-1} + 1$
Indri	2.2

Table 8: Weights for each IR sub-system used in the Hybrid Approaches

6.2.1 Hybrid Approach

Our third run (*UTDHLTCIR*) is our hybrid approach, based on Indri and both Lucene approaches, that uses NegEx to classify negations like our previous runs. This run mainly exists to provide comparisons between our second and fourth runs: to evaluate the effectiveness of the hybrid approach to the Lucene approaches.

6.2.2 Hybrid Approach with LingScope

Our final run (*UTDHLTCIRLS*) uses the hybrid approach but uses LingScope (Agarwal and Yu, 2010)[11] to detect negations and hedging within the documents. LingScope uses a Conditional Random Field trained on the BioScope (Szarvas et al., 2008) corpus of medical and clinical texts with annotated negations and speculations.

7 Result Filtering

After retrieval, results are filtered through three stages. First, documents are filtered to ensure that they satisfy any age requirements. This is accomplished by parsing the de-identified age information from each visit's reports and lowering the score for any visit that has more occurrences of de-identified ages outside its required range than within its required range. Next, if the query contains a gender requirement, visits are filtered to ensure that the reports associated with that visit contain more occurrences of words indicating the required gender than the opposing gender. Lastly, results are processed for negations. A visit is filtered if, for any required, non-negated keyword, more than one-third of the keyword's occurrences are negated (in the case of run 4, each hedged occurrence is counted as half of a negative occurrence). Likewise, a visit is also filtered if, for any required, negated keyword, less than one-third of the keyword's occurrences are negated (again, in the case of run 4, negated occurrences include hedged occurrences as half an occurrence).

[11] For our purposes, we use the provided *crf_scope_words_all_both* models for negation and hedge detection.

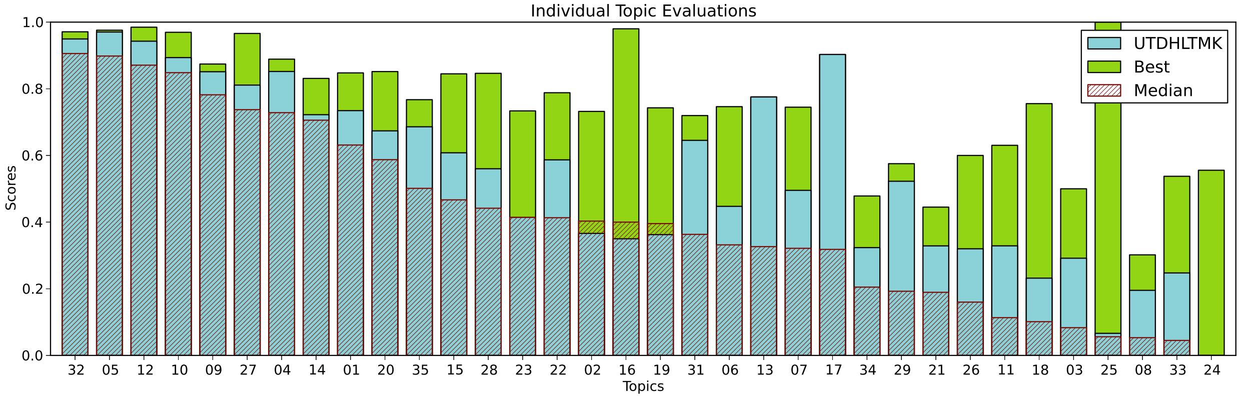

Figure 3: BPref for our system compared to the best and median scores for each topic.

8 Performance Evaluation

Table 9 summarizes the scores provided by NIST for our four submissions. While these measurements can be used to compare our submissions against each other, it is more prudent to compare our systems against those of the other groups. Figure 3 contains the binary preference scores for our second submission – the multi-keyword Lucene approach – to the best and median scores from all systems for each topic. It can be seen that our system generally performs between the top scoring system and the median.

Submission	**P10**	**MAP**	**BPref**
UTDHLTSL	56.2%	35.6%	50.8%
UTDHLTMK	63.2%	40.2%	54.3%
UTDHLTCIR	60.3%	40.8%	54.5%
UTDHLTCIRLS	60.0%	40.0%	53.4%

Table 9: Performance evaluations; *P10* refers to the percentage of relevant documents in the first ten results, *MAP* refers to the mean average precision, and *BPref* refers to the binary preference (Buckley and Voorhees, 2004).

Table 10 lists the effectiveness of each component in our system on relevance compared to that of our hybrid approach. Because only a small number of questions involved negation, age, or gender filtering, the effects of these components on our system is inconclusive. However, the effects of query expansion are clear: expanding keywords significantly improves the relevance of returned documents. Because all query expansion methods overlap (each provides a subset of the expansions generated by the other techniques), it is difficult to judge techniques individually. Despite this, it is clear that expansions based on co-occurrence information succeeded in locating documents which discuss relevant concepts. The results also suggest that SNOMED relations, UMLS, and Wikipedia provide essentially the same effects, and that future systems should be able to include any one of these techniques for comparable results.

Experiments	**MAP**	**Diff.**	**BPref**	**Diff.**
Negation				
NegEx	.4082	.0000	**.5449**	.0000
LingScope	.3995	-.0087	**.5339**	-.0110
None	.4165	.0083	**.5481**	.0032
Filtering				
W/o age filtering	.4107	.0025	**.5479**	.0030
W/o gender filtering	.4052	-.0030	**.5439**	-.0010
W/o any filtering	.4083	.0001	**.5476**	.0027
Query Expansion				
W/o co-occurrence	.3540	-.0542	**.4934**	-.0515
W/o SNOMED	.4085	.0003	**.5497**	.0048
W/o UMLS	.4034	-.0048	**.5435**	-.0014
W/o Wikipedia	.4036	-.0046	**.5451**	.0002
W/o any expansions	.3335	-.0747	**.4766**	-.0683
W/o any enhancements	.3301	-.0781	**.4778**	-.0671

Table 10: Experiments using the Hybrid Approach (UTDHLTCIR). All results are compared to that of the submitted system, which uses NegEx and all filtering and expansion enhancements.

9 Analysis

The problem of finding hospital visits corresponding to given patient cohorts, in principle, equates to finding sets of medical records centering on patients whom satisfy the traits of the given cohort. The subproblem of determining the relevance of each set of records requires extensive knowledge of the medical domain. We attempted to bridge this knowledge gap by using query expansion. Unfortunately, in some cases our expanded terms were too general and captured non-relevant records, and in other cases our terms were not general enough and caused us to miss records we should have considered relevant.

Topics twenty four and twenty five are particularly interesting in terms of potential improvements in that although we performed near the median, our performance was significantly lower than that of the highest performing system.

Our results for topic twenty four, *Patients who present to the hospital with episodes of acute loss of vision secondary to glaucoma*, are significantly damaged by incorrect keyword extraction. Rather than extracting the Wikipedia article title *loss of vision*, we extract *acute loss* and *vision* separately because OpenNLP splits the noun phrase on either side of the preposition *of*. Because the majority of Wikipedia article titles are noun phrases and because no automatic system has perfect accuracy, we could likely improve our results by scanning the entire topic for Wikipedia articles, rather than only scanning the topic's noun phrases.

Topic twenty five, *Patients co-infected with Hepatitis C and HIV*, presents a different problem. The "gold"-judged visits for that topic do not contain any discussion of Hepatitis C nor HIV. Instead, the gold visits contain ICD-9 codes indicative of Hepatitis C and HIV in their *discharge_diagnosis* field of their records. Although these records themselves contain no relevant discussion, the patient that they are associated to still satisfies the traits of the targeted cohort. This illustrates a subtlety in the task that we had not anticipated: rather than looking for visits that are relevant to the traits of the topic, we must instead ensure that the patient discussed satisfies the topic. Converting the keywords for each topic into their associated ICD-9 codes would have allowed our system address this subtlety by returning documents that lack keyword mentions.

In addition to these improvements, now that more training data is available, future implementations would likely find value in learning or inferring many of the parameters we set by intuition. Parameters of particular note are the weights for each level and rank of query expansion, and the threshold (ratio) used to reject visits containing too many incorrectly negated keyword occurrences as well as the weight of each hedged occurrence. Additionally, utilizing the section of an occurrence of a keyword would allow a system to assign more weight to related sections (e.g. "discharge summary") and less weight to potentially unrelated sections ("family history"). Likewise, this would allow a system to restrict the scope of a keyword to the relevant sections if one were actually looking for occurrences within a patient's family history.

10 Conclusions

The Medical Records Track was introduced in the 2011 Text REtrieval Conference. This track tackles the problem of collecting hospital visits that correspond to the traits of a given patient cohort. This gives rise to several difficulties: using collections of hospital records as the unit of response, working with highly complex medical texts for which each term requires a high degree of domain knowledge to comprehend, and the lack of training data imposed by the fact that is the first iteration of this track.

To participate in this task, our approach first extracts the requisite traits from each topic's cohort through keyword extraction, and then gender and age requirement extraction. Next, to bridge the domain knowledge gap, we expand each keyword through the UMLS and SNOMED medical databases, Wikipedia redirect information, and co-occurrence data gleaned from the PubMed Central Open Access Subset. Once we have expanded all of our keywords for a topic, we use the Lucene information retrieval engine (as well as Indri in two of our submissions) to initially rank our relevant visits. Finally, we prune these visits using age, gender, an negation filters.

Our results were promising: our best submission achieved a Mean Average Precision of of 40.8% and all of our submissions perform above the median for nearly all topics.

For future improvement, we plan to incorporate more medical-domain knowledge, such as ICD-9 codes, as well as using the judgements from this task to learn weights for our various parameters (query expansion weights, negation ratios, etc.) as well as incorporating section information in assessing the value of a keyword occurrence.

References

S. Agarwal and H. Yu. 2010. Detecting hedge cues and their scope in biomedical literature with conditional random fields. *Journal of biomedical informatics*.

O. Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267.

C. Buckley and E.M. Voorhees. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference* *on Research and development in information retrieval*, pages 25–32. ACM.

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

R.L. Cilibrasi and P.M.B. Vitanyi. 2007. The google similarity distance. *— IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pages 370–383.

M.C. De Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*. Citeseer.

C. Fellbaum. 1998. *WordNet: An electronic lexical database*. The MIT press.

E. Hatcher and O. Gospodnetic. 2005. *Lucene in Action*. Manning Publications.

Committee on Comparative Effective Research Prioritization and Institute of Medicine (US). 2009. *Initial national priorities for comparative effectiveness research*. National Academies Press.

M.Q. Stearns, C. Price, K.A. Spackman, and A.Y. Wang. 2001. Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association.

T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. 2004. Indri: A language model-based search engine for complex queries. In *Proceedings of the International conference on Intelligence Analysis*. Citeseer.

G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. 2008. The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45. Association for Computational Linguistics.