Search for Medical Records: NICTA at TREC 2011 Medical Track

Sarvnaz Karimi* David Martinez* Sumukh Ghodke*
Lumin Zhang† Hanna Suominen+ Lawrence Cavedon*
*NICTA and the University of Melbourne, Dept of CSSE, Australia
+NICTA and ANU, CECS, Australia
†Max Planck Institute for Biological Cybernetic, Germany
firstname.lastname@nicta.com.au

Abstract

NICTA (National ICT Australia) participated in the Medical Records track of TREC 2011 with seven automatic runs. The main techniques used in our submissions involved using Boolean retrieval for filtering, query transformation, and query expansion. Evaluation of our best run ranks our submissions higher than the median of all systems for this track, and stands at rank seven among 109 automatic runs which were submitted by the 29 participating groups.

1 Introduction

A medical record documents pertinent information concerning a given patients medical history, medical care, and current illnesses, typically across time and within a particular healthcare service providers jurisdiction.[1] This includes information needed for organizing, planning, performing, and controlling a good-quality medical care. The records consist of structured or numeric fields (approximately 60 per cent) and free text (approximately 40 per cent) [2]. More and more medical records are electronic.

Medical records allow healthcare professionals to provide appropriate and personalized care to their patients and customers. It is therefore important to develop searching capabilities that allow specific information to be extracted from a large number of records. The Medical Records track at TREC challenged participants with search in the free-text fields of electronic medical records. The test document collection for the Medical Records track was a set of de-identified medical records made available through the University of Pittsburgh BLULab NLP Repository. Reports associated with the same patient were identified and marked with unique visit identifiers. Participants received four sample queries, and 35 test queries. For the official evaluations, one of the test queries was omitted.

The NICTA team participated in the Medical Records track with three runs in the first round for pooling, and four official final runs. The main techniques used in our submission system were Boolean retrieval for filtering, query transformation, and query expansion. Details on these techniques are provided in the following sections. Throughout this report, depending on the context, we use "*document*" to refer to either each medical report (that is, an individual report in a medical record) or all reports related to a given visit.

[1] MeSH Medical Subject Headings 2011: http://www.nlm.nih.gov/mesh/MBrowser.html

2 Method

2.1 Document Collection Pre-processing

Prior to indexing, we processed all reports in the collection for normalization and document expansion, as described below.

- The medical records that were provided for this track contained rich metadata on diagnostics associated with each report. We expanded mentions of ICD9 codes[2] of admission and discharge diagnoses in the metadata with their descriptions as provided in ICD9 documents. Both the original code and expanded forms were included for indexing.

- Sections of the report texts were extracted using their headings or hand-crafted pattern-matching rules. To answer some of the queries, for example, we required to separate patients' *history* from their *current condition*. All reports were therefore processed to extract history (including *patient history*, *past medical history*, *past surgical history*), chief complaint, medications, allergies, admission and discharge diagnoses, and the full-text of the report. We used this information to create fields for Boolean search. The list of all such fields is given in Table 1.

- Some of the demographic information, such as gender, age, and specific conditions, such as patients weight, were only mentioned in the text. We used pattern matching to extract and normalize this information. For example, for gender, we replaced mentions of

  $$\boxed{\textit{gentleman, man, male, yo m, y/o M, boy}W}$$

  with *gendermale*, and mentions of

  $$\boxed{\textit{lady, woman, female, wm, girl}W}$$

  with *genderfemale*.

- Documents were processed to identify negated terms. Identifying negations reduced false positive matches against query terms. For example, if a query asks for patients *with chronic back pain* documents containing *no chronic back pain* will match. To resolve this, we created a list of negated words by running *NegEx*[3] over the entire collection. *NegEx* finds trigger terms (one word or multi-word expressions) that are clinical conditions negated in texts. Given a single term may have both positive and negative implications, we ignored negations that have more positive occurrences than negative within a report. All negated terms were replaced with a single word, no space, with a "*no*" prefix: e.g., if negation is implied for "chronic back pain", all instances of "chronic back pain" and its variants are replaced with the word "nochronicbackpain". This replacement is performed in the history, past history, present illness history, report text, medications, and allergies fields.

2.2 Query Processing

We used two levels of query processing to transform the bag-of-words queries into Boolean queries, and expand medical terms, as demonstrated below.

[2]International Statistical Classification of Diseases and Related Health Problems: `http://en.wikipedia.org/wiki/List_of_ICD-9_codes`

[3]`http://code.google.com/p/negex/`

Field	Description
ADMITDIAG	diagnostics during admission
AGE	patients age by decades, for example age30 means people in their thirties
ALLERGIES	allergies listed in the report
CHIEFCOMP	chief complaint, this may be equal to diagnostics during admission
DISCHDIAG	discharge diagnostics
GENDER	patient's gender extracted from text and represented as gendermale and genderfemale
HISTORY	history of the patient's medical condition or past medical illness
MEDICATIONS	medications
PRESTHIS	present illness medical history
PASTHIS	past medical history
REPORT	all the free text information, including history, past and present, and allergies

Table 1: List of fields defined for Boolean search.

What	**Pattern**	**Translation**
Gender	women/female	GENDER:gendermale
	men/male	GENDER:gendermale
Age	young adult	AGE:(age20 age30 age40)
	younger/young	AGE:(agebirth12 ageteen age20 age30 age40)
	adult	AGE:(age20 age30 age40 age50 age60 age70 age80 age90)
Weight		
Treatments	taking X (who|with|without|treated)	MEDICATIONS:X
	who are on X	MEDICATIONS:X
	patients on X for Y	MEDICATIONS:X
Admission	admitted (for|with) X who	CHIEFCOMP:X OR ADMITDIAG:X
Diagnostics	treated for X (who|during|while)	PRESTHIS:X OR DISCHDIAG:X
	(patients with|men with|women with) X	PRESTHIS:X OR DISCHDIAG:X
	who were discharged X	DISCHDIAG:X
History	with a* history of X (who|now)	HISTORY:X
Allergy	with X allergy	ALLERGY:X
	without allergy	ALLERGY:(noallergies)
Abbreviation	seen in the er|presented to the er	REPORT:("emergency room" OR ER)

Table 2: Rules (patterns in the queries and their translations) used in the query transformation step. Words that are all in capital letters are field names.

Query Transformation

We developed a set of manually constructed patterns to map query terms into the available fields (Table 1), translating them to the language of reports, or to expand them. These patterns — formed based on the sample clinical questions provided by the National Library of Medicine (NLM) [3] — covered seven broad categories of age, weight (using body mass index), diagnostics, treatments, medications, history, allergies, and abbreviations. For example, if a query contained "elderly patients", we expanded "elderly" with an equivalent age field that covered people in their 60s to 90+. Table 2 shows the details of the selected transformation rules. For example the query:

Elderly patients with ventilator-associated pneumonia

is translated to:

> *PRESTHIS:(ventilator associated pneumonia) OR DISCHDIAG:(ventilator associated pneumonia) OR AGE:(age60 age70 age80 age90) OR REPORT:(elderly with ventilator associated pneumonia).*

A small number of abbreviations, such as ER (emergency room), were also expanded in the queries.[4]

[4] A full list of all the transformation rules can be found in: `http://http://ww2.cs.mu.oz.au/~skarimi/codes/TRECMED.2011.qprocessing.pl`

Query Expansion

We leveraged external resources to add new terms to our queries, by identifying terms that are strongly related to the query terms. Specifically, we focused on query terms that represent medical categorical concepts (e.g. disease categories). For example, for the query below, we added terms falling under the category of "atypical antipsychotics":

Patients taking atypical antipsychotics

Our approach to expansion used two main knowledge sources: the UMLS Metathesaurus (version 2010AA) and DBpedia. In order to select expansion candidates we used *MetaMap-2010* from the National Library of Medicine (NLM). We defined manual expansion rules from these resources based on the sample queries and 50 priority queries from the NLM priority list.

For our final expansion system, we first applied MetaMap to identify phrases linked to terms in the UMLS Metathesaurus. The matched concepts were then used as candidate terms to be expanded; in some cases terms consisted of a primary term followed by a parenthesized description — such as "Intervention (Surgical and medical procedures)" — and in such cases we treated them as separate candidate terms.

Each candidate term had a Semantic Type (ST) associated with it in the MetaMap output. We used STs to define two expansion groups: safe expansion (for terms which STs include the string "Pharmacologic Substance") and filtered expansion (for terms whose ST is "Therapeutic or Preventive Procedure"). Candidate terms that did not belong to these groups were discarded. For the rest, if they were listed as "category" in DBpedia[5], we extracted all of the terms listed under the category as our expansion terms. For "safe expansion" the output was the full list of expansion terms; for "filtered expansion", we removed terms which are not UMLS concepts by applying MetaMap to each term.

In our implementation, we defined a small set of stop-categories that would have otherwise produced undesirable expansions. The following terms were excluded from expansion: "administration", "AMA", "diagnosis", "drug", "functional concept", "medication", and "surgery". We also removed terms with the following strings from the DBpedia output: "code", "history", "mechanism", "poisoning", "toxicity", and "withdrawal".

During the development process, we also explored expansion using hierarchical relations from the UMLS Metathesaurus; however, we observed that DBpedia offered a higher coverage of some domains, such as newly developed drugs, and less risk of over-expansion. For instance, one sample query contained the term "atypical antipshychotic", which UMLS expanded with 8 more specific drugs (e.g. "Clozapine"). DBpedia, however, identified the same set of drugs and as well as a further 22 new drug and brand names, which seemed correct after manual analysis, and had a stronger presence in the collection.

2.3 Indexing and Searching

We used two types of indexing in our runs: *visit-based* and *report-based*. In the former approach, all related reports for a visit were concatenated (removing duplicate diagnostics codes) to create a single "multi-document" item for indexing.

We used stop-word removal both in query processing and indexing; however, we augmented the typical list of stop-words with *patient*, and removed single characters, *and*, *or*, *not*, and *no* from the list.

The search engine used for indexing and searching in our runs was Apache Lucene (v3.2); we used both the BM25 and *tf-idf* ranking algorithms for Lucene [4]. We relied on field search in most our runs, i.e., a Boolean search followed by ranking. When Boolean search was turned off, the queries contents still did include field names but no Boolean operator. No stemming was done in our submitted runs.

[5] http://wiki.dbpedia.org/OnlineAccess

	Submitted Runs						
	NICTA1	NICTA2	NICTA3	NICTA4	NICTA5	NICTA6	NICTA7
transformation	✓		✓		✓	✓	
expansion		✓	✓	✓	✓	✓	✓
negation				✓	✓	✓	✓
Boolean	✓	✓	✓	✓	✓		
ranking function	BM25	BM25	BM25	tf-idf	tf-idf	tf-idf	tf-idf
indexing	V	V	V	V	R	R	R
stemming	none	none	none	none	none	none	none

Table 3: Specifications of the three initial runs (NICTA1 to NICTA3), and four official runs (NICTA4 to NICTA7). In *indexing*, V represents visit-based, and R means report-based.

	Submitted Runs						
Metric	NICTA1	NICTA2	NICTA3	NICTA4	NICTA5	NICTA6	NICTA7
Bpref	0.392	0.381	0.398	0.413	0.463	**0.490**	0.451
P@10	0.471	0.426	0.450	0.359	0.432	**0.503**	0.459
R-Prec	0.278	0.273	0.292	0.255	0.295	**0.355**	0.326

Table 4: Evaluation of the three initial runs (NICTA1 to NICTA3), and four official runs (NICTA4 to NICTA7).

3 Evaluation

We submitted three automatic runs for the first round of submissions, to be used for pooling; we later submitted a further four automatic runs as official runs. Specifications of these runs are summarized in Table 3. For the first three submissions we used Boolean search followed by BM25 ranking, and visit-based indexing. The official runs were a mix: report-based and visit-based indexing; Boolean search turned on and off; and the ranking function used was *tf-idf* rather than BM25. For all runs, we used query transformation and expansion, by themselves or combined.

The TREC organizers chose three metrics for evaluations: precision at 10 documents retrieved (P@10), R-precision (R-Prec), and Bpref [1]. The results for our submissions are summarised in Table 4. For the first three submissions (NICTA1 to NICTA3), our highest P@10 was achieved using NICTA1, which used only query transformation. However, the combination of query transformation and expansion gained higher Bpref and R-Prec results. Overall, we achieved consistently highest results, for all measures, using the NICTA6 configuration, which again used both steps of query processing — transformation followed by expansion — but with indexing at the report level; documents were processed for negation; and *tf-idf* was used for ranking. The queries were, however, flattened by removing the field names introduced in the query transformation step, leaving only their contents in the query.

Table 5 compares our best run, NICTA6, to the results of: the ideal best of submitted runs to TREC; the median of all submitted runs; and best run of the team which ranked top among all other submissions. TREC separated the results into *judged* — runs that participated and judged in the pooling — and *unjudged*, and reported the best results and median of those submissions. *Best* in the judged and unjudged categories was an artificial run where the best result of the individual queries were taken from all the submitted runs and formed an *ideal* run. Our best run is above the median for all three measures, and ranked seventh among 109 automatic runs submitted for

		Judged		Unjudged		
Metric	NICTA6	Best	Median	Best	Median	Best Run
Bpref	0.490	0.761	0.412	0.758	0.434	0.552
P@10	0.503	0.876	0.476	0.859	0.444	0.656
R-Prec	0.355	0.610	0.309	0.598	0.305	0.440

Table 5: Comparison of our best run (NICTA6) with the ideal best and median of 47 judged and 80 unjudged submissions, and best reported run (last column) among 109 automatic submissions.

	Complementary Runs						
	NICTA8	NICTA9	NICTA10	NICTA11	NICTA12	NICTA13	NICTA14
transformation	✓	✓	✓	✓	✓	✓	✓
expansion	✓	✓	✓	✓	✓	✓	✓
negation	✓	✓	✓	✓	✓	✓	
Boolean ranking function	BM25	BM25	BM25	BM25	tf-idf	tf-idf	tf-idf
indexing	V	V	R	R	R	V	R
stemming	none	Porter	none	Porter	Porter	Porter	Porter

Table 6: Specifications of the complementary runs which were not submitted to TREC. In *indexing*, V represents visit-based, and R means report-based.

	Complementary Runs						
Metric	NICTA8	NICTA9	NICTA10	NICTA11	NICTA12	NICTA13	NICTA14
Bpref	0.495	0.498	0.491	0.498	0.500	**0.508**	0.506
P@10	0.500	0.485	0.494	0.491	0.506	**0.535**	0.535
R-Prec	0.376	0.360	0.336	0.343	0.353	**0.377**	0.376

Table 7: Evaluation of the complementary.

this track (more information on the rankings of all teams is given in the track overview paper [5]).

Not all the possible combinations of search parameters — such as ranking function, visit-based indexing or report-based indexing — could be submitted to TREC. For completeness and to investigate changing which parameters help with the task, we ran all other missing configurations. We report some of the best configurations in our system in Tables 6 and 7. In all our submissions we had turned off stemming, which did hurt the effectiveness. The positive effect of adding stemming can be clearly seen in Table 7, when we used Porter stemmer. Using BM25 as a ranking function did not outperform *tf-idf* in any of the configurations. Our best performing run that was not submitted was NICTA13, which used stemming, visit-based indexing, *tf-idf* ranking, query transformation and expansion on the collection that was pre-processed to accommodate fields and handled negations (see Table 1).

4 Conclusions

The NICTA team submitted seven runs to the Medical Track of TREC 2011. We experimented with and submitted a variety of query processing and document processing approaches to this track. All documents were pre-processed to find — and, in specific cases, modify — concepts of interest. Among our submitted runs, the best results were achieved using query transformation — that is, breaking the query into different components and mapping these to their uniform representation as used in the documents — and query expansion using external resources.

Acknowledgments

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme.

Hanna Suominen was partially funded by the Academy of Finland (decision no. 136653). This work was performed while Lumin Zhang was a visiting student at the NICTA Canberra Research Laboratory.

References

[1] C. Buckley and E. M. Voorhees. 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, Sheffield, UK, 2004.

[2] H. Dalianis, M. Hassel, , and S. Velupillai. The stockholm EPR corpus – characteristics and some initial findings. In *The 14th International Symposium for Health Information Management Research*, pages 14–16, Kalmar, Sweden, 2009.

[3] Institute of Medicine. 100 initial priority topics for comparative effectiveness research, 2009.

[4] J. Perez-Iglesias, J. Perez-Aguera, V. Fresno, and Y. Feinstein. Integrating the probabilistic models BM25/BM25F into Lucene. *CoRR*, abs/0911.5046, 2009.

[5] E. M. Voorhees and R. M. Tong. Overview of the TREC 2011 medical records track. In *The tenth Text REtrieval Conference*, Gaithersburg, MD. National Institute of Standards and Technology.