DutchHatTrick: Semantic query modeling, ConText, section detection, and match score maximization.

Martijn Schuemie[*]
ErasmusMC
m.schuemie@erasmusmc.nl

Dolf Trieschnigg[†]
University of Twente
trieschn@ewi.utwente.nl

Edgar Meij[‡]
University of Amsterdam
edgar.meij@uva.nl

Introduction

This report discusses the collaborative work of the ErasmusMC, University of Twente, and the University of Amsterdam on the TREC 2011 Medical track. Here, the task is to retrieve patient visits from the University of Pittsburgh NLP Repository for 35 topics. The repository consists of 101,711 patient reports, and a patient visit was recorded in one or more reports.

Because the training set provided by the track organization was small and not made available until quite late in the competition, we decided to create a small training set ourselves. Not only did this allow us to test several ideas before submitting runs to TREC, it also led to several insights into the data. One finding was that synonyms are widely used. Query expansion was therefore deemed essential to achieve a reasonable performance. Query expansion has been used before in Information Retrieval (IR), and is often divided into statistical and knowledge-based query expansion. Statistical query expansion uses data derived from the corpus itself, and a well-known example is pseudo-relevance feedback [1]. In contrast, we investigated knowledge-based query expansion, which uses a knowledge base such as an ontology or a dictionary to find related terms. This type of query expansion has not always proven to be successful. For instance, Hersh et al. [2] found a decrease in overall search performance when using the Unified Medical Language System (UMLS) [3] to find related terms. Liu et al. [4] found slight improvements with scenario-specific expansion strategies using UMLS. In a previous TREC track, we also found reduced performance when using concept based query expansion [5], but found slightly improved results when using an approach combining concepts with a statistical model of related words [6]. Similarly, Zhou [7] found promising results when using combination of both the original words in the text and the synonyms found for concepts in the text.

An often-used resource for knowledge-based query expansion in the biomedical domain is the UMLS. However, initial explorations indicated that there is only limited overlap between terms used in topics and medical records and terms found in the UMLS. The main reason for this appears to be that the UMLS is mainly constructed from vocabularies used in classifying clinical data, but not intended to be used in text-mining. Terms in the UMLS tend to be more specific than what a physician would use in free-text reporting. For instance, a physician might use the term ‘upper endoscopy’, but this term is not found in the UMLS. Instead, the term ‘upper GI endoscopy’ is found. We have therefore explored a different source of synonyms: Wikipedia. We expected Wikipedia to have a better coverage of the terms encountered in medical records.

[*] IPCI group, Medical Informatics Department, Erasmus University Medical Center of Rotterdam, the Netherlands
[†] Database Group, University of Twente, Enschede, The Netherlands
[‡] Information and Language Processing Systems group, Intelligent Systems Lab, University of Amsterdam, the Netherlands

In the following section the construction of the training set is discussed in detail, followed by a description of our system, and an overview of its performance.

Construction of a training corpus

Epidemiologists at the Erasmus University Medical Center were asked to name several topics relating diseases to treatments representing interesting epidemiological research questions that could be investigated using hospital medical records. Five topics were generated, and these were combined with the example topic provided by the TREC organization (topic 0) as shown in Table 1. For each topic, the original query was manually expanded with terms proposed by the epidemiologists in an effort to maximize recall.

ID	Query	Manual query expansion
0	gastroesophageal reflux disease AND upper endoscopy	(gastroesophageal reflux disease OR reflux) AND ((upper AND endoscopy) OR esophagogastroduodenoscopy OR EGD OR OGD OR Oesophagogastroduodenoscopy)
1	subarachnoid bleeding AND clipping	((subarachnoid AND bleeding) OR (subarachnoid AND hemorrhage) OR (subarachnoid AND haemorrhage) OR SAH) AND (clipping OR closure by clip OR Elgiloy OR Sugita)
2	subarachnoid bleeding AND coiling	((subarachnoid AND bleeding) OR (subarachnoid AND hemorrhage) OR (subarachnoid AND haemorrhage) OR SAH) AND (coiling OR Guglielmi OR GDC)
3	Guillain Barre syndrome AND immunoglobulin	(Guillain Barre syndrome OR GBS OR Landry's paralysis) AND (immunoglobulin OR IVIg OR IgG)
4	trigeminal neuralgia AND microvascular decompression	(trigeminal neuralgia OR tic douloureux OR prosopalgia OR Fothergill's disease) AND (microvascular decompression OR MVD OR Jannetta procedure)
5	upper GI bleed AND past NSAID use	(stomach bleeding OR (upper AND (gastrointestinal OR GI) AND (hemorrhage OR bleed OR bleeding))) AND (NSAID OR NSAIDs OR Aspirin OR Aceclofenac OR Acemetacin OR Alclofenac OR Alminoprofen OR Azapropazone OR Benoxaprofen OR Benzydamine OR Bufexamac OR Bumadizone OR Chondroitin OR Clofezone OR Dexibuprofen OR Dexketoprofen OR Diacerein OR Diclofenac OR DiclofenacPiroxicam OR Difenpiramide OR Droxicam OR Etodolac OR Etoricoxib OR Fenbufen OR Fenoprofen OR Fentiazac OR Feprazone OR Flufenamic OR Flunoxaprofen OR Flurbiprofen OR Glucosamine OR Glucosaminoglycan OR Ibuprofen OR Ibuproxam OR Indometacin OR Indoprofen OR Kebuzone OR Ketoprofen OR Ketorolac OR Lonazolac OR Lornoxicam OR Lumiracoxib OR Meclofenamic OR MeloxicamIbuprofen OR Mofebutazone OR Morniflumate OR Naproxcinod OR Naproxen OR Naproxen and esomeprazole OR Niflumic OR Nimesulide OR Orgotein OR Oxaceprol OR Oxametacin OR Oxaprozin OR Oxyphenbutazone OR Parecoxib OR Phenylbutazone OR Pirprofen OR Proglumetacin OR Proquazone OR Rofecoxib OR Sulindac OR Suprofen OR Tenidap OR Tenoxicam OR Tiaprofenic acid OR Tolfenamic OR Tolmetin OR Valdecoxib OR Zomepirac)

Table 1. Set of topics created in collaboration with epidemiologists. The manual query expansion was performed to include as many synonyms as possible.

A Lucene [8] instance containing the reports in the corpus was created. Using the UMLS, ICD-9 codes in the admit_diagnosis and discharge_diagnosis fields were expanded with their text label before indexation by Lucene to allow reports to be found on these ICD-9 codes. For each manually expanded query a set of

reports was retrieved using Lucene. For topics 0, 4, and 5, the top 100 ranking reports were used for further analysis. Topics 1, 2, and 3 returned only 45, 70, and 64 reports respectively, and all these reports were used. The reports were manually classified as Relevent, Irrelevant, or Questionable. Questionable reports were removed from the set, and were not condidered in any of the subsequent analysis.

Lessons learned during manual classification

The manual classification of records was in itself an informative exercise, providing several insights into the data:

- Synonyms are commonly used. For instance, a topic might refer to "upper endoscopy", but most physicians will use the abbreviation "EGD" instead.
- ICD-9 codes were found to be uninformative and appear to be used only to indicate what was screened for during a procedure, rather than indicate a final diagnosis. For example, in report19244.xml the ICD-9 code 530.81 (Gastro-esophageal reflux disease) is included in the discharge_diagnosis field, but the report states that "The esophagus was normal… There was no evidence of stricture, inflammation or ulcers," which is in direct contradiction with the finding of gastroesophageal reflux disease.
- Some sections are more informative than others; we found that sections labeled "postoperative diagnosis" are the most reliable source for locating diseases.
- A topic might mention a drug class, such as all NSAIDs, but reports will often mention only an individual drug belonging to the class, such as Aspirin.

Information retrieval methods

Preprocessing corpus

The text in the original reports was chopped into lines with a maximum length, probably to accommodate display in the hospital information system. This splitting had to be undone to form complete sentences, which was needed by some of our text processing. First, the number of characters at which sentences were cut off had to be identified, and this cutoff varied per record (67-80 characters). The maximum line length in a report after correcting for insertion of tags by anonymizing was used as the cutoff length for that report. If a line plus the first word in the next line exceeded this cutoff length, this was assumed to indicate that splitting had taken place, and the lines were concatenated.

Next, the text was split into sentences. For this the OpenNLP sentence boundary detection algorithm [9] was used.

A simple spelling correction algorithm was applied to the text: Two sets of words were extracted from the corpus. The first set consisted of probably correctly spelled words, containing all words that either appeared frequently in the corpus (n >= 50), or were found in a vocabulary of known words, either directly or after lemmatization. The vocabulary of known words comprised all words in UMLS (2010AB) and Wordnet [10]. The second set consisted of probably misspelled words, which were words that appeared infrequently (n < 50) and were not found in the vocabulary of known words. Misspelled words were mapped to correctly spelled words if the Levenshtein distance equaled one. If multiple candidate correct words were found, the most frequent word was taken. Examples of mappings are: 'fragnents' (n =

1) to ‘fragments’ (n = 2,279), ‘stenosispresent’ (n = 1) to ‘stenosis present’ (n = 12,110), and ‘possiblity’ (n = 16) to ‘possibility’ (n = 3,282). In total, 10,252 words were corrected in the corpus.

The ConText algorithm [11] was used to identify parts of sentences that were negated (e.g. ‘rule out pneumonia’) , referred to someone else than the patient (e.g. ‘family history of pneumonia’), or occurred in the past (e.g. ‘past history of pneumonia’). ConText is a simple algorithm, similar to the NegEx algorithm [12], that uses trigger terms and simple rules to determine whether a part of a sentence belongs to one of the three categories mentioned above. For example, the trigger term ‘rule out’ indicates that the remainder of the sentence is negated.

Section detection

A simple rule-based system was used to split the report into sections. In a first pass the report is scanned line-by-line to detect the start and endings of a section. In a second pass additional section starts and ends are added to make sure the sections do not nest or overlap. Where possible a label is assigned to the section. The rules to detect the start and endings of sections were manually created based on an analysis of a representative sample from the collection, i.e. containing examples of different report types.

Typically, a line consisting of a sequence of uppercase letters, optionally followed by a colon, marked the start of a section. In some reports the actual content was written in uppercase letters as well. A number of heuristics (for instance, too many words in the title, the existence of an explicit line end such as a question mark or period, or the start of the line looking like an enumeration) were used to prevent flagging these lines as section starts. Lines containing only series of hyphens were used to detect section ends.

Table 2 shows the 15 most frequently extracted section titles from the report collection.

Label	Frequency
impression	47,900
findings	42,477
examination performed	37,275
technique	27,557
physical examination	22,572
allergies	21,936
history of present illness	21,341
review of systems	21,007
social history	19,827
medications	18,232
abdomen	15,804
heent	15,558
chief complaint	13,998
extremities	9,389
vital signs	8,728

Table 2. Most frequently found section labels.

Even though non of the sections could be completely ignored, there were some sections that provided stronger evidence for the relevance of search terms, and some sections where a mention of a search term were considered less informative. Table 3 shows the manually created list of strong and weak section headers.

Strong sections	Weak sections
chief complaint	admission diagnosis
diagnosis	clinical history
discharge diagnoses	comparison
discharge diagnosis	family history
discharge summary	institution
final diagnoses	name
final diagnosis	past medical history
post op diagnosis	pre op diagnosis
postoperative diagnosis	preoperative diagnosis

Table 3. Manually identified strong and weak sections.

Preprocessing queries

The sample topics provided by the TREC organization showed that the format of topics was not a clean boolean query as used in our

own training set. Instead, full natural language sentences were provided such as 'Patients taking atypical antipsychotics without a diagnosis schizophrenia or bipolar depression'. Non-relevant terms in the query were identified as either belong to a small list of predetermined stop words (all stopwords used in MEDLINE plus 'patients', 'who', 'or', 'admission', 'discharge', 'hospital'), or were part of either a verb phrase or prepositional phrase as identified using the shallow parsing algorithm of the OpenNLP toolkit [9]. Parts of queries that were identified by the NegEx algorithm [12] as being negated were considered to be required not be present in relevant documents. However, negations did neither appear in our own training set, nor in the TREC test set.

Finding synonyms using Wikipedia

Wikipedia provides a rich and extensive source of information, not only in terms of content but also more structural information, such as the hyperlinks between articles. Besides the general domain, the amount and quality of medical information in Wikipedia also continues to grow. For our current work, we leverage the anchor texts of incoming hyperlinks to Wikipedia articles, including not only "normal" hyperlinks, but also redirects and alternative titles of pages. In particular we calculate the *link probability* of an anchor text as well as the *prior probability* for a given anchor text with respect to an article. The former is a function of an n-gram and determines the probability that it can be a link to a Wikipedia article. The latter is a function of an n-gram and an article and indicates the probability that the n-gram is used as a textual representation for that article.

Finding synonyms and hyponyms using Drugbank and UMLS

Initial investigation of using Wikipedia to relate drug classes to individual drugs showed that it is not trivial to detect such hierarchical relationships accurately in the Wikipedia source files. Instead, a combination of UMLS and DrugBank [13] was used. DrugBank is an extensive and accurate repository of drug names and synonyms, but it does not contain a structured representation of drug classes. DrugBank entries were mapped to UMLS concepts using exact string matching of drug names, and parent concepts were identified using the 'is-a' relationship in the UMLS. Names and synonyms of the parent concepts were extracted from the UMLS.

If a query contained the name of a drug class, the query was expanded with the synonyms found for that drug class, and the names and synonyms of all the drugs in the class.

Semantic Query Modeling

Query likelihood (QL) determines the probability that a document generated the query and ranks the documents accordingly. We use a maximum likelihood estimate (MLE) on the query and calculate QL according to the KL-divergence of the MLE query and each document language model. SQM builds upon this by updating the query model using information derived from Wikipedia.

Semantic Query Modeling using Wikipedia

We implemented a novel algorithm that uses anchor information in a principled fashion. It does so in two steps. First, the anchor texts are used to identify and score relevant Wikipedia articles for all possible term n-grams in a query. Then, for each of these articles, we again use the incoming anchor texts. In this step, however, we use them to determine the parameters of a language model for each article. The language models are subsequently combined for all n-grams in the query, yielding as end result a semantically-

informed language model of the query, i.e., a semantic query model (SQM). For retrieval we use a standard KL-divergence approach, with Bayesian smoothing using a Dirchlet prior.

Match score maximization

One potential problem of query expansion is that some aspects of the query are expanded with many synonyms, whereas other aspects are expanded only with a few or none. This can create an imbalance in aspect representation (query drift) when matching a query to documents. One solution to this problem is to allow each part of a query to be linked to at most one term in a document. When a query part can be linked, for instance through expansion with synonyms, to many terms in the document, the term giving the maximum partial matching score is selected.

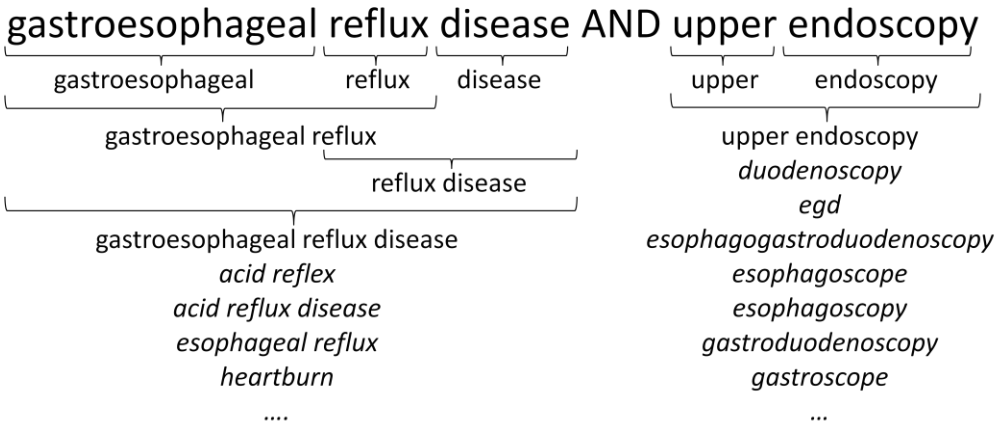

Figure 1. Query terms after expansion, and their relation to parts of the original query. Terms in italics are terms added by the query expansion. 'AND' is in the list of stopwords, and is therefore ignored.

Figure 1 shows an example of query terms after expansion, and how these terms relate to parts of the original query. The original query terms are also maintained. If, for instance, a document was matched to the query because it contained the term 'egd', no additional score would be added to the match score if the term 'upper' was also found in the document.

A term could be matched to a document in four ways: *exact matching* indicates all words in the term were found in the correct order with no words in between. *Sentence*, *section* and *report matching* indicates all words were found, but separately, within a single sentence, section or report, respectively. Term and document frequencies for each type of matching were calculated separately.

The partial matching score for each query term was calculated using the Okapi BM25 model

```
if term is orginal query term then weight = 1
else if term is child of drug class then weight = 0.1
else if term is Wikipedia synonym  then {
  if term is not redirect and term linktag occurrence < 3 then weight = 0
  else weight = link probability * prior probability
}

if term found in strong section then weight = weight * 5
else if term found in weak section then weight = 0
```

Figure 2. Pseudocode of the algorithm for calculating the weight applied to the term frequency.

[14], with two modifications: First, a very large bonus (1,000,000) was added to the score for every character of the query linked to the matched term. For example, in figure 1, if the term ‘egd’ was found, 15 characters of the query would be linked, adding 15,000,000 to the score. The purpose of this was to give priority to documents that matched a larger portion of the query, and is in line with earlier observations that IR systems tend to fail because not all aspects of a topic are present in the returned documents [15]. Second, the term frequency is weighted by term characteristics, using the algorithm shown in Figure 2.

The algorithm in figure 2 has several parameters that needed to be selected. Also, for the Okapi model there are the *b* and *k* parameters. These parameters were optimized using a crude broad search on the training set. The optimal values are shown in figure 2, and the optimal values for *b* and *k* are 0.75 and 2, respectively. The performance did not vary much with the selection of parameters: the minimum MAP encountered was 0.74.

Runs

Several matching algorithms were tested, each one using the preprocessing steps described in this paper. All algorithms searched for relevant reports instead of visits, and the matching score for a visit was calculated as the maximum matching score of the reports belonging to that visit.

Lucene baseline (baselucene)

As a baseline, a Lucene instance was created with the default Lucene settings. Parts of sentences identified by the ConText algorithm as being negated, being related to someone other than the patient, or referring to the past were held out from the index. Information on the section headers was not used. In this baseline run, no query expansion was applied.

Lucene with query expansion (explucene)

A second run utilized the same Lucene instance as the baseline run, but queries were expanded using both Wikipedia and the DrugBank-UMLS combination. Synonyms from Wikipedia were added if the term was a Wikipedia redirect or the anchor text of an incoming hyperlink that occurs at least five times.

SQM (SQM)

This run uses the same collection preprocessing steps as the “Lucene with query expansion” run and performs retrieval using the SQM method defined above.

Match score maximization (WWOCorrect)

This run uses the same collection preprocessing steps as the “Lucene with query expansion” run and performs retrieval using the “Match Score Maximization” method defined above.

Results

	AP0	AP1	AP2	AP3	AP4	AP5	**MAP**
Lucene baseline	0.72	0.77	0.73	0.67	0.88	0.13	**0.63**
Lucene, ConText filtering*	0.72	0.78	0.79	0.47	0.86	0.12	**0.61**
Lucene, manual query expansion	0.73	0.55	0.63	0.67	0.82	0.16	**0.60**
Lucene, manual query expansion, ConText filtering	0.78	0.59	0.59	0.67	0.82	0.33	**0.64**
Lucene, auto query expansion	0.70	0.80	0.76	0.67	0.86	0.11	**0.63**
Lucene, auto query expansion, ConText filtering*	0.73	0.80	0.80	0.67	0.87	0.13	**0.65**
QL	0.67	0.82	0.82	1.00	0.96	0.16	**0.71**
SQM*	0.57	0.62	0.82	0.77	0.95	0.55	**0.71**
Match score maximization*	0.59	0.88	0.76	1.00	0.96	0.49	**0.75**

Table 4. Results of various runs on our own training set. The Average Precision (AP) is shown per topic, as well as the Mean Average Precision (MAP). Colors indicate precision; red indicates the lowest precision, green indicates the highest precision. * these runs are comparable to the submitted runs.

Table 4 shows the results of the various matching strategies on our own training set. In contrast to the official tests, these evaluations were performed at the report level, not at the visit level. The results indicate that a combination of query expansion and ConText filtering improves performance, and that the match score maximization is able to achieve the highest score. One should keep in mind that the system was developed based on this training set, and that results are probably positively biased.

Table 5 shows the official results on the TREC test data. None of the advanced matching strategies is significantly better than the Lucene baseline run using ConText filtering (based on a one-sided paired sample t-test, $p < 0.05$). On the contrary, the Lucene baseline significantly outperforms SQM and match score maximization at P10.

	Lucene, ConText filtering (baselucene)			Lucene, auto query expansion, ConText filtering* (explucene)			SQM* (SQM)			Match score maximization (WWOCorrect)		
Topic	BPREF	R-prec	P10	BPREF	R-prec	P10	BPREF	R-prec	P10	BPREF	R-prec	P10
101	0.70	0.45	1.0	0.80	0.57	1.0	0.78	0.51	0.7	0.85	0.38	0.4
102	0.28	0.22	0.6	0.33	0.22	0.6	0.48	0.24	0.5	0.45	0.13	0.3
103	0.05	0.08	0.1	0.08	0.08	0.1	0.50	0.58	0.6	0.43	0.33	0.4
104	0.67	0.67	0.6	0.73	0.67	0.6	0.84	0.89	0.8	0.56	0.56	0.5
105	0.90	0.50	0.8	0.90	0.47	1.0	0.90	0.46	1.0	0.92	0.32	0.2
106	0.20	0.13	0.1	0.20	0.12	0.2	0.33	0.18	0.5	0.45	0.14	0.2
107	0.23	0.26	0.5	0.29	0.30	0.6	0.16	0.26	0.2	0.32	0.30	0.6
108	0.25	0.31	0.4	0.28	0.31	0.4	0.08	0.08	0.1	0.14	0.15	0.1
109	0.72	0.37	0.9	0.73	0.38	0.9	0.18	0.04	0.0	0.82	0.14	0.1
110	0.92	0.57	1.0	0.92	0.58	1.0	0.91	0.51	1.0	0.85	0.37	0.6
111	0.38	0.38	0.5	0.25	0.29	0.4	0.09	0.10	0.2	0.23	0.05	0.1
112	0.70	0.34	0.7	0.71	0.41	0.7	0.69	0.40	0.9	0.85	0.62	1.0
113	0.09	0.07	0.1	0.39	0.43	0.4	0.31	0.29	0.2	0.54	0.57	0.5
114	0.76	0.53	0.9	0.78	0.58	0.9	0.75	0.45	0.6	0.84	0.55	0.8
115	0.50	0.47	0.5	0.50	0.47	0.4	0.51	0.47	0.4	0.55	0.31	0.4
116	0.68	0.70	0.7	0.00	0.00	0.0	0.78	0.80	0.8	0.64	0.70	0.7
117	0.22	0.14	0.3	0.18	0.18	0.3	0.17	0.09	0.1	0.61	0.59	0.6
118	0.57	0.40	0.9	0.57	0.40	0.9	0.10	0.10	0.4	0.32	0.13	0.3
119	0.49	0.37	0.9	0.53	0.35	0.7	0.24	0.24	0.1	0.44	0.26	0.6
120	0.64	0.42	0.7	0.59	0.41	0.6	0.28	0.08	0.0	0.50	0.27	0.8
121	0.36	0.23	0.3	0.26	0.18	0.1	0.41	0.45	0.6	0.43	0.30	0.2
122	0.61	0.58	0.6	0.65	0.67	0.7	0.79	0.46	0.8	0.58	0.29	0.3
123	0.48	0.55	0.6	0.46	0.48	0.5	0.39	0.27	0.6	0.51	0.58	0.7
124	0.03	0.17	0.2	0.44	0.50	0.3	0.06	0.17	0.1	0.00	0.00	0.0
125	0.00	0.00	0.0	0.04	0.07	0.1	0.00	0.00	0.0	0.00	0.00	0.0
126	0.00	0.00	0.1	0.00	0.00	0.2	0.24	0.40	0.3	0.12	0.20	0.4
127	0.84	0.47	0.8	0.73	0.44	0.8	0.50	0.15	0.3	0.50	0.21	0.2
128	0.44	0.21	0.4	0.47	0.24	0.5	0.38	0.14	0.4	0.37	0.09	0.0
129	0.46	0.38	0.8	0.51	0.40	0.8	0.49	0.42	0.7	0.46	0.43	0.5
131	0.36	0.20	0.6	0.36	0.20	0.4	0.36	0.20	0.7	0.58	0.22	0.8
132	0.91	0.63	1.0	0.94	0.68	1.0	0.92	0.64	1.0	0.92	0.66	0.7
133	0.11	0.10	0.1	0.14	0.15	0.2	0.27	0.20	0.4	0.06	0.00	0.0
134	0.20	0.26	0.3	0.21	0.21	0.4	0.10	0.18	0.1	0.24	0.12	0.2
135	0.64	0.56	0.9	0.58	0.39	0.6	0.57	0.32	0.4	0.73	0.44	0.9
all	0.45	0.34	0.56	0.46	0.35	0.54	0.43	0.32	0.46	0.49	0.31	0.41

Table 5. Official results on the TREC test set. Colors indicate score; red indicates worst scores, green indicate best scores. * These runs were judged.

Conclusions

Although the results on the training data looked promising, the official evaluation shows that knowledge-based query expansion remains problematic. The use of Wikipedia instead of more traditional knowledge resources appears not to give a robust improvement in search performance.

The filtering of parts of sentences that are negated, refer to someone other than the patient, or refer to historical events, gives a higher performance on the training set when combined with either manual or automatic query expansion. However, we did not include a run in the official test where this filtering was not applied.

References

1. Buckley, C., et al., *Automatic Query Expansion Using SMART*, in *Proceedings of TREC 3*. 1994.
2. Hersh, W., S. Price, and L. Donohoe, *Assessing thesaurus-based query expansion using the UMLS Metathesaurus.* Proc AMIA Symp, 2000: p. 344-8.
3. Bodenreider, O., *The Unified Medical Language System (UMLS): integrating biomedical terminology.* Nucleic Acids Res, 2004. **32**(Database issue): p. D267-70.
4. Liu, Z. and W. Chu, *Knowledge-based query expansion to support scenario-specific retrieval of medical free text.* Information Retrieval, 2007. **10**(2): p. 173-202.
5. Trieschnigg, D., W. Kraaij, and M.J. Schuemie, *Concept Based Document Retrieval for Genomics Literature*, in *Proceedings of TREC 2006*. 2006.
6. Schuemie, M.J., D. Trieschnigg, and W. Kraaij, *Cross Language Information Retrieval for Biomedical Literature*, in *Proceedings of TREC 2007*. 2007.
7. Zhou, W., et al., *Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature*, in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 2007, ACM: Amsterdam, The Netherlands. p. 655-662.
8. Gospodnetic, O. and E. Hatcher, *Lucene in Action*. 2005, Greenwich: Manning Publications.
9. *OpenNLP*. Available from: http://incubator.apache.org/opennlp/.
10. Miller, G.A., *WordNet: A Lexical Database for English.* Communications of the ACM, 1995. **38**(11): p. 39-41.
11. Harkema, H., et al., *ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports.* Journal of Biomedical Informatics, 2009. **42**(5): p. 839-851.
12. Chapman, W.W., et al., *A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries.* Journal of Biomedical Informatics, 2001. **34**(5): p. 301-310.
13. Knox, C., et al., *DrugBank 3.0: a comprehensive resource for 'omics' research on drugs.* Nucleic Acids Res, 2011. **39**(Database issue): p. D1035-41.
14. Robertson, S.E., S. Walker, and M. Hancock-Beaulieu, *Okapi at TREC- 7: Automatic Ad Hoc, Filtering, VLC and Interactive Track*, in *Proceedings of the Seventh Text REtrieval Conference*. 1998: Gaithersburg, MD, USA. p. 199–210.
15. Buckley, C., *Why current IR engines fail*, in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 2004, ACM: Sheffield, United Kingdom. p. 584-585.