UCLA at TREC 2014 Clinical Decision Support Track: Exploring Language Models, Query Expansion, and Boosting

Jean I. Garcia-Gathright[a], Frank Meng[a,b], William Hsu[a,b]

University of California, Los Angeles

[a] *Department of Bioengineering*
[b] *Department of Radiological Sciences*

Abstract

For the TREC 2014 Clinical Decision Support track, participants were given a set of 30 patient cases in the form of a short natural language description and a data set of over 700,000 full-text articles from PubMed Central. The task was to retrieve articles relevant to the patient cases and one of three types of clinical question: diagnosis, test, and treatment.

This paper describes the retrieval system developed by the Medical Imaging Informatics group at the University of California, Los Angeles. One manual run and four automatic runs were submitted. For the automatic runs, a variety of retrieval strategies were explored. Two retrieval methods were compared: the vector space model with TF-IDF similarity, and a unigram language model with Jelinek-Mercer smoothing. The performance of retrieving on abstracts alone was compared to that of full-text. Finally, a simple set of rules for query expansion and term boosting was developed based on recommendations from domain experts.

Submissions for 26 groups were pooled and evaluated by a team of medical librarians and physicians at the National Institute of Standards and Technology. The results showed that 1) the language model outperformed the vector space model for automatically-constructed queries, 2) searching full-text was more effective than searching abstracts alone, and 3) boosting improved the ranking of retrieved documents for "test" topics, but not "diagnosis" topics. Our best automatic run used the language model, full-text search, query expansion, and no boosting.

1. Introduction

PubMed, the National Library of Medicine's database of over 23 million citations, is an important source of knowledge in the biomedical domain. In order to leverage PubMed to provide evidence-based decision support to clinicians at the point-of-care, it is necessary to develop strategies to retrieve literature relevant to specific patients. Retrieved documents should address common clinical questions such as: what is the patient's diagnosis? What tests should the patient receive? What treatment should the patient receive?

The TREC 2014 Clinical Decision Support track challenged participants to retrieve articles from PubMed Central, an open-access subset of PubMed, given a short description of a patient case and an associated clinical question. 30 patient cases were provided, as well as a data set of over 700,000 full-text articles from PubMed Central. This paper describes the retrieval system developed by the Medical Imaging Informatics group at the University of California, Los Angeles.

Preprint submitted to TREC 2014 Proceedings *January 29, 2015*

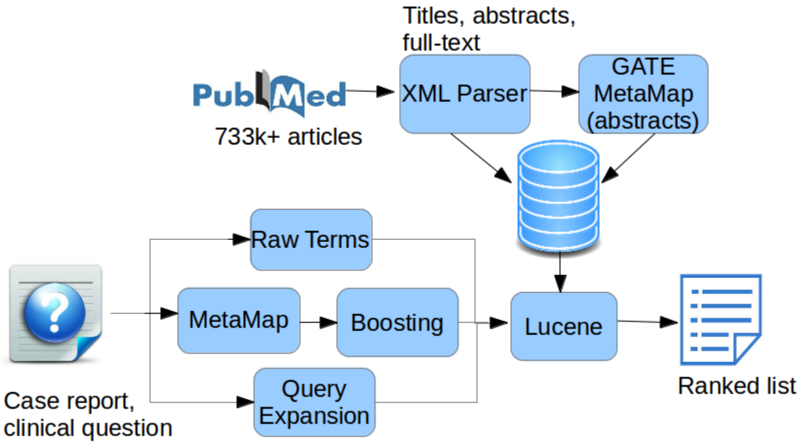


Figure 1: Overview of retrieval system.

2. Methods

2.1. Indexing

The PubMed Central articles were given as a set of XML files, one file per article. An XML parser, implemented in Python, extracted PMC IDs, keywords, titles, abstracts, and full-texts from the XML. If an abstract was not available for an article, the Conclusion section of the text, extracted with a regular expression, was used as a substitute for the abstract.

The GATE MetaMap plugin [1,2] mapped the abstracts of the articles to the Unified Medical Language System (UMLS) [3], a controlled vocabulary for biomedical terminology. The mapping included identification of concept names, concept codes, semantic types, negation status, and match scores. To prevent excessive numbers of matched concepts, only match scores greater than 800 were used.

Data for each article was stored in a relational database. Two indexes were created using Apache Lucene [4]. The first index used Lucene's default scoring method: a vector space model using TF-IDF weighting and cosine similarity. The second index was produced using a language modeling approach, in which documents are represented as a unigram language model, then ranked based on the likelihood of the model generating the given query. Language model approaches require a smoothing method to account for terms that appear in the query but not in the document. We chose to use Jelinek-Mercer smoothing, with a lambda of 0.7, as recommended in [5] for retrieving articles from lengthy queries.

2.2. Recommendations from domain experts

Two clinicians were recruited to provide recommendations for designing and improving the query construction process. The first clinician composed ad-hoc queries based on the patient description and clinical question. For many topics, the clinician was able to recall the correct diagnosis from memory. However, we chose to assume that for "diagnosis" topics, diagnosis is part of the information need and would not be known at the time of the query. Thus, for "diagnosis" questions, the clinician was instructed not to include the diagnosis explicitly in the query. For test and treatment topics, no such constraint was enforced. These expert queries were used to retrieve a baseline set of results, submitted to TREC as a manual run.

A second clinician reviewed the first pass of retrieval results, produced using mostly keyword-based searches. The clinician described her approach to understanding the patient cases and suggested modifications to the retrieval strategy.

The two medical experts agreed independently in several aspects. First, they emphasized the importance of capturing patient attributes such as age and sex, as these contribute important information to the diagnostic process. For example, pediatric cases, women of childbearing age, and geriatic populations each have a distinct set of possible medical complaints. The experts also recommended discriminating between acute and chronic or progressive conditions, as well as including a term that specifically names the clinical question type (e.g., *diagnosis, test, testing, treatment, management*). Both experts relied on domain knowledge to rank the discriminative value of symptoms and recognize patient features that point to probable diagnoses.

2.3. Query construction

The automatic query construction algorithm was designed to approximate recommendations from domain experts. Base queries were produced from the condensed patient summaries. Patient summaries were mapped to UMLS codes using MetaMap. The base query consisted of the patient summary itself, concatenated with the list of UMLS concept codes. Concepts identified by MetaMap as being negated were not included in the query. Two types of phrases were found to cause false positive matches. "Xx-year old" caused many of the top matches to be case reports; sometimes, the only matching aspect of the report was the patient's age. The word "her" also caused articles on HER, a driver mutation in breast cancer, to be retrieved. Both of these special cases were removed from the base query.

Query expansion was used for three purposes: to focus the query on the clinical question type, to highlight the temporal aspect of the patient's condition, and to retrieve articles on pediatric care as needed. First, the query was expanded based on the clinical question associated with the case. For a test-oriented query, added terms included *test, evaluate, diagnose, guideline, examination, measurement, imaging*, and UMLS semantic types *Diagnostic Procedure* and *Laboratory Procedure*. Likewise, to retrieve treatment-related articles, additional terms were *treat, treatment, manage, management, therapy, guidelines, intervention*, and semantic types *Therapeutic or Preventative Procedure* and *Pharmacologic Substance*.

Query expansion was also used to characterize the temporal condition of the patient and whether the case described a pediatric patient. The patient summary was searched for time-related terms. If the words *hours* or *days* were found, the query was expanded to include the word *acute*. If the words *months* or *years* appeared, the query included the word *chronic*. Similarly, if the UMLS mapping of the patient summary identified the patient as a child, the term *pediatric* was added to the query.

Certain query terms were boosted if they belonged to a highly relevant semantic class. *Signs or Symptoms, Findings, Diseases and Syndromes, Injuries or Poisonings, Therapeutic or Preventative Procedures*, and *Pharmacologic Substances* were boosted by a factor of 20. *Population Groups* and *Age Groups* were boosted by a factor of 50. Boost factors were chosen empirically, based on the importance of the semantic type and the perceived qualitative benefit to the retrieved results compared to lower and higher boost factors.

Query terms	Comments
right lower quadrant abdominal pain, decreased appetite enlarged appendix abdominal ultrasound.	Raw terms, stopwords removed, Xx-year-old phrase removed
metamap_concept:Entire appendix metamap_concept:Transabdominal Ultrasound	UMLS mapping, no boosting
metamap_concept:Female child ^50 metamap_concept:Right lower quadrant pain ^20 metamap_concept:Decrease in appetite ^20	Boosting population group and symptoms
+(treat treatment manage management therapy guideline intervention) pediatric	Query expansion for topic type and pediatric case

Table 1: Query terms for patient summary: *15-year-old girl with right lower quadrant abdominal pain for hours, decreased appetite, and enlarged appendix on abdominal ultrasound.*

Table 1 describes the query terms for an example patient summary. The final query is the concatenation of the list of query terms.

One manual run and four automatic runs were submitted, each exploring a dimension of the retrieval strategy. The runs were:

1. **tfman**: manual run using expert queries, vector space model with TF-IDF weighting, and full-text search.
2. **tfauto**: automatic run using vector space model with TF-IDF weighting, query expansion, and full-text search.
3. **jmab**: automatic run using language model with Jelinek-Mercer smoothing, query expansion, and abstracts only.
4. **jmignore:** automatic run using language model with Jelinek-Mercer smoothing, query expansion, and full-text search.
5. **jmboost**: automatic run using language model with Jelinek-Mercer smoothing, query expansion, boosting, and full-text search.

tfman, produced by expert query, provided a baseline for retrieval performance. For the automatic runs, we hypothesized that 1) the language model would outperform the vector space model, 2) full-text search would be more effective than searching abstracts alone, and 3) boosting would improve results vs. no boosting.

2.4. Evaluation

26 groups participated in the TREC Clinical Decision Support track, submitting 91 automatic runs and 11 manual runs. The judgement set consisted of documents ranked 1-20 in any runs, union a 20% sample of documents ranked 21-200 in some run. Documents were judged as being (potentially) relevant, or not relevant. Evaluation metrics were: inferred average precision (infAP); inferred normalized discounted cumulative gain (infNDCG), which measures how well the documents were ranked; precision at R (R-prec), where R is the number of known relevant documents; and precision at 10 (P@10).

Evaluation metrics were calculated for each topic. For each run, the average of each metric was calculated over all topics. Averages were also calculated after stratifying topics by clinical question type. Statistical significance was determined by pairwise approximate randomization.

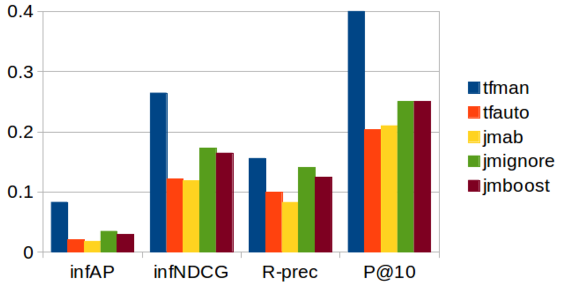


Figure 2: Comparison of all submitted runs. **tfman** was the best run overall. **jmignore** was the best automatic run.

	infAP	**infNDCG**	**R-prec**	**P@10**
tfauto	**0.026**	**0.003**	**0.002**	0.141
jmab	**< 0.001**	**< 0.001**	**< 0.001**	0.121
jmboost	0.220	0.451	**0.032**	0.972

Table 2: P-values for comparisons of automatic runs to **jmignore**. Bold indicates statistical significance ($p < 0.05$).

3. Results

3.1. Comparing submitted runs to each other

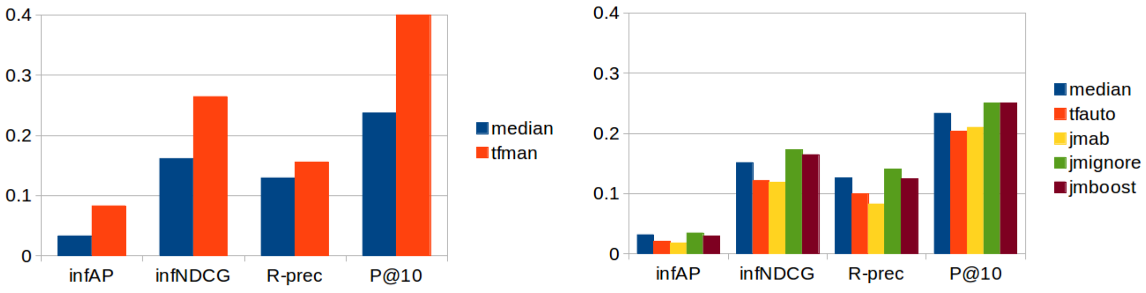
Figure 2 shows each run's results for each metric, averaged over all topics. Table 2 summarizes significance levels for this comparison set. As expected, **tfman** had the best results for each metric. **tfauto** and **jmab** did not perform as well as **jmignore**, confirming our hypotheses that the language model and full-text search were more effective strategies. **jmignore** performed slightly better than **jmboost**, although the difference was significant for R-prec only. Contrary to our hypothesis, boosting did not improve retrieval when averaged over all topics.

3.2. Comparing submitted runs to the median

Figure 3 and Table 3 compare our manual and automatic runs with their respective medians over all TREC CDS participants. For the automatic runs, **tfauto** and **jmab** performed below the median. **jmboost** performed slightly below the median for infAP and R-prec, and slightly above the median for infNDCG and P@10, although these differences were not statistically significant. Our best automatic run, **jmignore**, performed slightly above the median for all metrics (again, not statistically significant). **tfman** was our best run overall, performing in the top 1-2 over all TREC CDS participants for all metrics except R-prec.

3.3. Analyzing best automatic runs by topic type

Figure 4 breaks down our best automatic runs by topic type and compares them to the median. Figure 4a shows that **jmignore** had the best performance for "diagnosis" topics, reaching statistical significance over **jmboost** but not over the median. Thus, boosting was detrimental to retrieval for "diagnosis" topics.



(a)

(b)

Figure 3: Comparison of submitted runs to the median. (a) **tfman** performed well above the median. (b) **jmignore** was slightly above the median, but was not statistically significant.

	infAP	**infNDCG**	**R-prec**	**P@10**
tfman	**0.020**	**0.014**	0.288	**0.002**
tfauto	0.319	0.064	0.076	0.316
jmab	0.227	0.132	**0.032**	0.575
jmignore	0.718	0.143	0.258	0.547
jmboost	0.855	0.365	0.878	0.546

Table 3: P-values for comparisons of all submitted runs to the median. Bold: $p < 0.05$. Underline: $p < 0.10$.

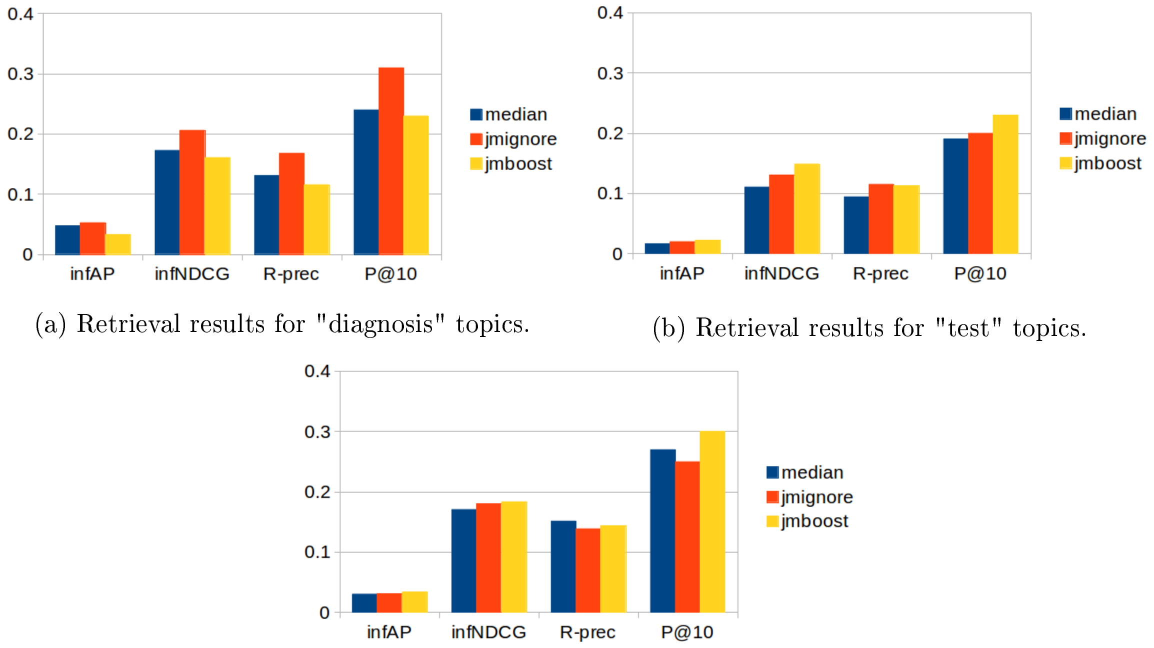
Figure 4b summarizes performance for "test" topics: for this set of patient cases, **jmboost** provided some benefit over the median ($p < 0.1$). As seen in Figure 4c, **jmboost** improved P@10 for "treatment" topics, but this finding did not reach statistical significance.

Table 4 summarizes the p-values for each comparison set, by topic type.

4. Discussion

tfman was an excellent run overall, demonstrating the value of expert knowledge in domain-specific systems. Two of our three hypotheses were confirmed: for the automatic runs, the language model approach outperformed the vector space model. These results are consistent with our preliminary evaluations as well as previous work showing performance gains from the language model approach over TF-IDF weighting [6]. Full-text search performed significantly better than searching abstracts alone, demonstrating the importance of using full-text in applications when available. One surprising result was that semantic boosting did not improve retrieval overall. However, boosting did improve retrieval for some metrics, particularly on "test" topics.

An investigation of the best- and worst-performing boost queries revealed a few trends. One observation was that short queries containing relatively little information benefited most from boosting. Because "test" topics generally contained less information than "diagnosis" topics, this may explain the improved performance on "test" topics with boosting. For example, this "test" topics performed well: *85-year-old* ***man*** *who was in a car accident 3 weeks ago, now with 3 days of progressively decreasing level of* ***consciousness*** *and impaired ability to perform*



(a) Retrieval results for "diagnosis" topics.

(b) Retrieval results for "test" topics.

(c) Retrieval results for "treatment" topics.

Figure 4: Comparison of best automatic runs by topic type. **jmboost** improves ranking for "test" topics, but not "diagnosis" topics.

	infAP	**infNDCG**	**R-prec**	**P@10**
jmignore vs. median				
diagnosis	0.846	0.304	0.284	0.382
test	0.331	0.329	0.094	1.0
treatment	0.967	0.652	0.515	0.773
jmboost vs. median				
diagnosis	0.713	0.746	0.586	1.0
test	0.076	0.054	0.062	0.313
treatment	0.926	0.637	0.699	0.633
jmboost vs. **jmignore**				
diagnosis	**0.013**	**0.026**	**0.005**	**0.032**
test	0.251	0.100	0.714	0.521
treatment	0.655	0.800	0.574	0.496

Table 4: P-values for comparisons of **jmignore**, **jmboost** and median, broken down by topic type. Bold: $p < 0.05$. Underline: $p < 0.10$.

activities of daily living. Boosting of UMLS concept "consciousness-related finding" brought relevant articles to the top of the result list compared to no boosting.

However, boosting introduced further noise in this lengthy "diagnosis" topic: *2-year-old* ***boy*** *with* ***fever*** *and irritability for 5 days. Physical exam findings include* ***conjunctivitis, strawberry tongue****, and* ***desquamation*** *of the fingers and toes. Lab results include* ***low albumin****, elevated white blood cell count and C-reactive protein, and* ***urine leukocytes****. Echo shows moderate dilation of the coronary arteries.*

Secondly, performance was sensitive to errors in selection of boosting terms. Consider the following "test" topic (bold indicates boosted terms): *25-year-old* ***woman*** *with* ***fatigue, hair loss, weight gain****, and* ***cold intolerance*** *for 6 months.* Median performance for this topic was relatively low, despite being an easy diagnosis of hypothyroidism for a medical expert. However, boosting performed well compared to no boosting. Because this topic has very little noise and all relevant terms were boosted, an article on hypothyroidism was ranked 3rd in the retrieval results. Without boosting, the highest-ranked article on hypothyroidism was ranked 16th.

In contrast, this "diagnosis" topic performed poorly: *67-year-old* ***woman*** *status post* ***cardiac catheterization*** *via right femoral artery, now with a cool, pulseless right foot and right* ***femoral bruit****.* In this instance, the relevant term "femoral artery" was not boosted, resulting in retrieval of documents only generally related to cardiac catheterization.

Performance with boosting also suffered when less-relevant terms were boosted. For example, boosting of "hypertension" and "obesity" skewed the result list in this "diagnosis" topic: *58-year-old* ***woman*** *with* ***hypertension*** *and* ***obesity*** *presents with exercise-related episodic* ***chest pain radiating*** *to the back.*

These observations suggest that a well-tuned boosting strategy could improve the ranking of documents relevant to specific patient cases. One potential area of development would be in calculating the boost weight for each term. In our system, all boosted semantic types receive one of two boost values (20 or 50). However, our expert queries reflect that symptoms have a range of discriminative power and this knowledge is used by experts to produce short, dense queries. Consider the topic *8-year-old boy with 2 days of loose stools, fever, and cough after returning from a trip to Colorado. Chest x-ray shows bilateral lung infiltrates.* The expert query, *pediatric pulmonary infection colorado*, integrated the concepts *fever* and *cough* into a single query term, *pulmonary infection*, whereas the concept *bilateral lung infiltrates* was not included in the query at all. A knowledge- or data-driven scoring function that estimates the discriminative power of symptoms could be an area of further investigation.

5. Conclusion

This paper described the retrieval methods developed by the UCLA Medical Imaging Informatics group for the TREC 2014 Clinical Decision Support shared task. A suite of methods that included a unigram language model with Jelinek-Mercer smoothing, query expansion, and full-text search performed near the median. For a subset of the topics, a boosting strategy based on semantic type conferred some benefit to the ranking of retrieved documents, suggesting that with further development it could be a viable strategy for the matching of literature to patient cases.

Acknowledgements

This work was supported by NLM T15-LM007356, NIH/NLM R01-LM009961, and the UCLA Department of Radiological Sciences.

References

1. Cunningham H, Tablan V, Roberts A, Bontcheva K. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLoS Comput Biol 2013; 9(2): e1002854.

2. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association 2010; 17(3): 229-236.

3. Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research 2004; 32(suppl 1): D267-D270.

4. McCandless M, Hatcher E, Gospodnetic O. Lucene in Action: Covers Apache Lucene 3.0. Manning Publications Co.; 2010.

5. Zhai C, Lafferty J. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) 2014; 22(2): 179-214.

6. Ponte JM, Croft WB. A language modeling approach to information retrieval. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. 1998; 275-281.