Medical Case-based Retrieval by Leveraging Medical Ontology and Physician Feedback: UIUC-IBM at ImageCLEF 2010

Parikshit Sondhi[1], Jimeng Sun[2], ChengXiang Zhai[1], Robert Sorrentino[2], Martin S. Kohn[2], Shahram Ebadollahi[2], Yanen Li[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
{sondhi1,czhai,yanenli2}@illinois.edu, {jimeng,sorrentino,marty.kohn,ebad}@us.ibm.com

Abstract: This paper reports the experiment results of the UIUC-IBM team in participating in the medical case retrieval task of ImageCLEF 2010. We experimented with multiple methods to leverage medical ontology and user (physician) feedback; both have worked very well, achieving the best retrieval performance among all the submissions.

Keywords: Medical information retrieval, Language model, Medical ontology, MeSH, UMLS, Feedback

1 Introduction

The Text Information Management group at the University of Illinois at Urbana-Champaign and the Healthcare Transformation group at IBM TJ Watson Research Center collaborated in participating in the medical case retrieval task of ImageCLEF 2010. This paper is a report of our experiments and findings based on preliminary analysis of the results of our submissions.

The medical case retrieval task involved searching medical literature to find cases similar to a sample case specified in a query case. The query case provided a text description of a patient's background, symptoms and relevant test findings as well as a set of images such as CT scans. The following are text descriptions in two representative queries with different lengths:

Topic 17: "Female patient, 25 years old, with fatigue and a swallowing disorder (dysphagia worsening during a meal). The frontal chest X-ray shows opacity with clear contours in contact with the right heart border. Right hilar structures are visible through

the mass. The lateral X-ray confirms the presence of a mass in the anterior mediastinum. On CT images, the mass has a relatively homogeneous tissue density."

Topic 18: "Pain and incapacity to move after an accident. Slight deformation can be seen in the x-ray."

Their corresponding images are shown in Figure 1.

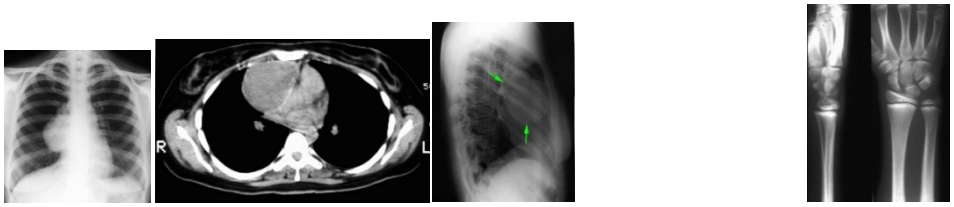

Fig. 1. Images for Topic 17 (left three) and Topic 18 (right one)

The document collection is a set of literature articles published in Radiology and Radiographics. Each article also includes the text of the captions and a link to the html of the full text articles. Images from these articles are also provided. The retrieval task is to run a case query on this data set to retrieve all the similar cases to the query case from this set of articles. For this task, a "case" is regarded as equivalent to an article that covers a medical case. Thus from computational perspective, we can simply treat each article as a unit and cast the task as one to rank all the articles based on a query case, much similar to an ordinary text retrieval problem. A ranked list of up to 1,000 articles (i.e., cases) can be submitted for each query case, which would then be evaluated using standard retrieval measures. More detailed descriptions of this task and its design can be found at the overview paper by the organizers [1].

Participants of this task in the past have found that although images are provided, matching cases solely based on text information seems to be not only sufficient, but also performs very well (see, e.g., [2]), thus we have focused on using only text information to perform medical case retrieval. Our goal of participation is two-fold: First, we would like to see how well a well-tuned state-of-the-art text retrieval model would work for this task. Second, we would like to see whether we can improve the state-of-the-art retrieval models by leveraging medical ontology and user (physician) feedback. Preliminary analysis of our experiment results shows that a standard retrieval model works reasonably well for this task, and both medical ontology and physician feedback can further improve retrieval accuracy over a standard state-of-the-art retrieval model.

In the rest of this paper, we first describe the retrieval methods we used in producing our runs, particularly how we leverage medical ontology and incorporate physician feedback, and then discuss our experiment results.

2 Basic Retrieval Methods

As our first line of experiments, we analyzed the performance of state of the art general retrieval methods for this task. This helped us assess the utility of such methods for the task and also obtain a strong baseline method for our further experiments with techniques to leverage medical ontology and physician feedback. These baseline experiments also allowed us to identify the differences between general retrieval and medical case retrieval and provide insights about how to address these differences by extending the standard state of the art retrieval methods. These extensions will be described in detail in the subsequent sections. For standard retrieval models, we used their implementations provided in the LEMUR retrieval toolkit (http://www.lemurproject.org/).

2.1 KL-divergence Retrieval Model with Dirichlet Smoothing

Language modeling provides a systematic framework for designing retrieval models. One of the best-performing retrieval models based on language modeling is the Kullback-Leibler (KL) divergence retrieval model [3]. Given a query Q and a document D, this model would first estimate a unigram query language model θ_Q (i.e., a word distribution) based on a given query and a document language model θ_D for document D, and then score the document D with respect to query Q based on negative KL-divergence between the two language models, $-D(\theta_Q \,||\theta_D)$, defined below:

$$-D(\theta_Q||\theta_D) = -\sum_{w \in V} p(w|\theta_Q)\, log\frac{p(w|\theta_Q)}{p(w|\theta_D)}$$

where V is the set of words in our vocabulary, and $p(w|\ \theta_Q)$ and $p(w|\ \theta_D)$ are the probabilities of word w given by the two language models, respectively. The negative KL-divergence intuitively measures the similarity of the query language model and the document language model, thus would favor a document with more query words.

The document language model $p(w|\ \theta_D)$ is usually estimated using Dirichlet prior smoothing, which often performs the best [3]:

$$p(w|\theta_D) = \frac{c(w,D) + \mu p(w|C)}{|D| + \mu}$$

Where $c(w,D)$ is the count of word w in document D, $p(w|C)$ is a background/reference language model estimated based on all the documents in the collection and helps providing probabilities for words unseen in a document, and μ is a smoothing parameter, which was tuned using last year's dataset with 5 queries. The optimal value was set at $\mu = 4800$.

The simplest way to estimate the query model θ_Q is to set $p(w|\ \theta_Q)$ to the relative frequency of a word in the query: $p(w|\theta_Q) = \frac{c(w,Q)}{|D|}$.

Since this approach assigns zero probability to words not in the query, a potentially better way to estimate this model is to use a technique called pseudo relevance feedback, which we discuss below.

2.2 Pseudo Relevance Feedback

Pseudo relevance feedback is a standard approach meant for improving retrieval performance via query expansion. It assumes top *N* documents in the ranked list generated by the baseline method as relevant and then picks a set of keywords from those documents and adds them to the query. Although not all the top-ranked documents are relevant, they do resemble relevant documents and often can suggest useful related terms to the query to expand and enrich a query representation. In general, such methods would pick terms that are far more common in the top-ranked documents in an initial retrieval result but not very common in the whole collection. With this strategy, we can estimate $p(w|\ \theta_Q)$ based on both the query and the top-ranked documents. In our experiments, we used the mixture model approach described in [4], which is one of the best-performing state of the art approaches to pseudo feedback. This approach is available in the Lemur toolkit that we used.

The mixture model pseudo feedback method has a few parameters, which we tuned using the 5 queries of last year's dataset. The best results were found when the number of documents used for feedback was set to top two documents.

Based on experiment results with last year's data set, we found that pseudo feedback generally improves performance over the simple relative frequency estimation method, though the improvement was largely variable. This is to be expected since with pseudo feedback, we simply blindly assumed the top ranked documents are relevant; in reality, these top-ranked documents are unlikely all relevant and may be distracting, thus hurting performance. Nevertheless, we decided to use the combination of Dirichlet prior smoothing with mixture model for pseudo feedback as our baseline method, which represents the best we could achieve with a state-of-the-art retrieval model out of the box from an existing retrieval toolkit with parameters tuned based on last year's data set. As we will discuss later, this baseline run actually worked very well.

3 Understanding the Challenges

From our experiments with the existing state of the art retrieval methods, we observed certain weaknesses that make them unsuitable for the case retrieval task. In this section, we discuss them in detail. In particular we focus on the unique characteristics that differentiate medical case retrieval from general retrieval.

We realized that the performance dropped for the following reasons:

Vocabulary Gap: Medical domain uses a highly specialized language, involving long multi-word expressions, term-order variations and abbreviations etc. It is often the case that, the same medical concept may have many different keyword variations. As a result, the keywords used in the query do not exactly match the conceptually similar, but morphologically different variants used in the documents. We term this as the vocabulary gap problem.

Non-Optimal Query Term Weighting: Case retrieval queries contain information regarding a patient's background, symptoms, any test results/observations etc. and are in general much longer than general search queries. As a result many of the query keywords are not very useful in identifying a case. The primary heuristic used in a standard retrieval model for judging the query keyword importance is based on the Inverse Document Frequency (IDF) (i.e., penalizing a word in the collection) does not seem to work well in this case. Based on this insight, we thus proposed to weigh keywords based on their semantic categories. The weight of keywords should also account for their semantic categories. Keywords belonging to certain categories like disease names (eg. cancer, diabetes etc.), symptoms (eg. dry cough, headache) etc. must be assigned high importance regardless of their IDF.

Missing Condition Names: Condition/disease names are usually the most discriminative keywords for finding similar cases. However such keywords representing *potential diagnoses* are often absent from the case descriptions provided as queries.

4 Leveraging Semantic Resources to Overcome the Challenges

Our subsequent experiments were aimed at overcoming the challenges described in the previous section. A major advantage in the biomedical domain is that a plethora of domain specific resources, such as the UMLS and MeSH ontologies, the MMTX toolkit etc. are available for language processing tasks. We believe that these can be used to address some of the limitations present in the general retrieval methods. We start by giving a brief description of these resources and then present our methods for leveraging them.

MeSH Terms: Medical Subject Headings (MeSH) is the U.S. National Library of Medicine's controlled vocabulary used for indexing articles for MEDLINE/PubMed (http://www.nlm.nih.gov/mesh/). Each indexed research article is assigned a set of representative MeSH terms. MeSH terminology provides a consistent way to retrieve information that may use different terminology for the same concepts. MeSH database can be used to find Medical Subject Heading Terms and build a search strategy.

Unified Medical Language System: NLM's Unified Medical Language System (UMLS) project develops and distributes multi-purpose, electronic "Knowledge Sources" and associated lexical programs for system developers and researchers

(http://www.nlm.nih.gov/research/umls/). They are useful in investigating knowledge representation and retrieval questions. UMLS contain three knowledge sources: Metathesaurus, Semantic Network and Specialist Lexicon. The main component is the Metathesaurus, which compiles and cross-references one hundred biomedical terminologies (in version 2003AA: more than 800,000 concepts and 2,000,000 strings), with their hierarchical and transversal relations. Its Semantic Network comprises 133 broad semantic groups and adds a common structure above these imported terminologies. The Specialist Lexicon provides a large English lexicon with an emphasis on biomedical words, including derivational knowledge. Tools have also been built around the UMLS to address terminological variation.

MMTx: MetaMap Transfer (MMTx) is a tool to perform the task of mapping UMLS biomedical concepts and semantic groups to free text (http://mmtx.nlm.nih.gov/). The biomedical concepts used for mapping are taken from the Unified Medical Language System. The system is also capable of identifying multi-word expressions, synonyms, abbreviations, term variants and stop words.

4.1 Keyword Weighing using UMLS Semantic Groups

In order to deal with the keyword weighing problem, we mapped the query keywords to UMLS semantic groups and then assigned weights based on the groups. Based on our analysis of all the semantic groups, we selected the following groups:

Disease or Syndrome, Body Part organ or organ component, Sign or Symptom, Finding, Acquired Abnormality, Congenital Abnormality, Mental or Behavioral Dysfunction, Neoplasm, Pharmacologic Substance

Query keywords belonging to each of these groups were assigned twice the weight of all other keywords. That is, their probabilities in the query language model are doubled. These groups were chosen specifically as most keywords belonging to these categories were found to be fairly discriminative while finding a relevant case.

4.2 MeSH based Pseudo-Relevance Feedback

We note that physicians tend to use the following strategy to decide if a document is relevant to a given case:

1. Look at the available patient background, symptoms and test results information
2. Make a list of possible conditions based on the available information
3. All documents discussing those conditions have a high probability of being relevant

This approach gives an important insight. Keywords representing potential conditions, which are completely missing from our queries, are highly useful in identifying similar

cases. Moreover, assuming the query case description is reasonably descriptive we can assume that there would only be a small number of conditions or potential diagnoses for that case. Thus if we can somehow guess these conditions and push up the documents that primarily talk about them, we should be able to improve performance.

This breaks down into two problems:

1. *We need a way of knowing which conditions a given document talks about:* Since each medical literature article indexed in Pubmed already has a set of MeSH terms assigned to it, we can easily filter out condition related MeSH terms to identify the prominent conditions the document talks about.
2. *We need some guess on what conditions the query case is likely to represent*: This is a harder problem and we deal with it in two ways. These are discussed in the following.

Top-N-based MeSH feedback

We make a list of all condition related MeSH terms present in the top N=10 documents in the initial ranked list generated by the baseline method. We then slightly reduce the weight of any documents below these top N that do not have any MeSH terms in common with this list. The method has an advantage in that it not only directly finds documents belonging to same conditions, but also it altogether avoids the problem of selecting and weighing the document keywords for pseudo relevance feedback. One limitation of the approach however is that we cannot re-rank the top N documents using it. We overcome this limitation in our second approach.

Distribution based MeSH feedback

Let M be the set of all condition related MeSH terms. Then for a given query Q, the method works as follows:

1. For every MeSH term m in M, set $Score(m) = 0$
2. Retrieve a ranked list of all the documents L for the query Q
3. For each document d in L
    a. Identify the set of query keywords S_d (subset of Q) found in d
    b. Identify the set of MeSH terms M_d (subset of M) found in d
    c. For each MeSH term m in M_d:
        i. For each query keyword q in S_d:
            1. If we have never encountered the keyword q in a document labeled with MeSH term m, then $Score(m) = Score(m) + 1$
4. Sort all MeSH terms m in M in descending order of $Score(m)$.
5. Select the top N=25 MeSH terms from the ranked list
6. Re-rank documents by reducing the weights of all documents not labeled with any of these N selected MeSH terms.

The intuition behind this method is that we assume that MeSH terms, whose documents contain a large number of query keywords, are more likely to represent the query. This approach is robust in that it takes into account the entire ranked list, rather than just the top ranked documents. It is therefore not affected by poor ranking generated by the baseline method. Additionally we are able to also re-rank the top results.

5 Approaches Utilizing Physician Feedback

Our next idea at dealing with the vocabulary gap problem was to let the doctors decide which keywords they considered most useful. Additionally what other related keywords they thought would be useful in detecting the right cases. This is to simulate an application scenario where the physician users would be able to use a search engine to reformulate the query with potentially more useful keywords.

We used these keywords to expand our queries. We assigned low weights to these keywords to prevent them from dominating the original keywords. We observed that this strategy helped improve performance across all queries. Additional keywords from doctors helped considerably in overcoming the vocabulary gap problem. In many cases the doctors provided condition/disease keywords representing potential diagnosis. This also helped greatly.

To further leverage feedback information from a user in an interactive retrieval system, we also experimented with relevance feedback, which is also a standard technique in improving search results. The idea is to ask a user to judge a small number of top-ranked documents as relevant or non-relevant, and the system could then use such judgments to improve the ranking of additional unseen documents for this user or improve ranking of all the documents for such a query case for future users. In our experiments, we asked two physicians to judge the top 20 documents and then used the same mixture model that we used for pseudo feedback to improve the estimation of query language model based on documents actually judged as relevant (as opposed to assuming top ranked documents to be relevant). However, our relevance feedback runs did not perform as well as we expected. This can be caused by multiple reasons, which we will further discuss later.

6 Summarizing the Submitted Results

In this section we give a brief description of our runs and provide a preliminary analysis of the results. We submitted 10 runs; the first four runs (run IDs: 1-4) are completely automatic, while the rest six runs (run IDs: 5-10) utilized an additional set of manually provided keywords for query expansion. In addition, the last three runs (run IDs: 8-10) also utilize relevance feedback provided by the users. We now describe these runs in more detail. The order (sequence number) is the same as the run IDs that we used to label our

runs (i.e., the first run described below is our submitted run with the run ID 1, and the second has the run ID 2, etc). The relations between these runs are shown in Figure 2.

1. **1276844704028__baselinefbsub:** This was a basic retrieval run. It used the KL-divergence retrieval model with Dirichlet prior smoothing and pseudo-relevance feedback as described in Section 2. The performance of this run gives us a sense of what can be achieved using a well-tuned existing state-of-the-art method.
2. **1276846614397__baselinefbWMR_10_0.2sub:** This run added to Run 1 two additional heuristics: UMLS based keyword weighing and top-N-based MeSH feedback. For MeSH based re-ranking as discussed in section 4.2, the weights are reduced by 0.2. Comparing this run with Run 1 would allow us to see whether keyword weighting and top-N-based MeSH feedback are indeed effective.
3. **1276846564056__baselinefbWMD_25_0.2sub:** This run is very similar to Run 2 except that we used the distribution-based MeSH feedback instead of the top-N-based MeSH feedback. For MeSH based re-ranking as discussed in section 4.2, the weights are reduced by 0.2. This run can be compared with Run 1 to see any improvement from keyword weighting and distribution-based MeSH feedback or compared with Run 2 to see which of the two MeSH feedback methods (i.e., top-N-based vs. distribution-based) works better.
4. **1276846825574__baselinefbWsub:** This run added to Run 1 only UMLS-based keyword reweighting. (In Runs 2 and 3, we added not only UMLS-based keyword reweighting, but also a MeSH feedback method.) Thus comparing this run with Run 1 can reveal any improvement from just using UMLS-based keyword reweighting, while comparing this run with Run 2 or Run 3 would allow us to see whether any of these MeSH feedback methods can further improve performance on top of keyword reweighting.
5. **1276848633547__PhybaselinefbWsub:** This run is similar to Run 4 but with additional keywords from physicians included. When compared with 4, this run lets us analyze the benefit from obtaining additional related query keywords from the users.
6. **1276849026093__PhybaselinefbWMR_10_0.2sub:** This run added, on top of Run 2, additional keywords from physicians.
7. **1276850977593__PhybaselinefbWMD_25_0.2sub:** This run added, on top of Run 3, additional keywords from physicians. Thus Runs 6 and 7 attempted to improve Runs 2 and 3, respectively, by adding additional keywords from physicians to further enrich the query. Meanwhile, comparing Run 6 and Run 7 can also further examine the relative effectiveness of the two MeSH feedback strategies.
8. **1276859628288__PhybaselineRelfbWMR_10_0.2sub:** This run added relevance feedback on top of Run 6. First, a set of results were generated using Run 6. A physician was then asked to identify the relevant documents among the top 20 results. The query was then re-executed using the same configuration as Run 6, but this time we used the documents labeled by the physicians for relevance feedback (instead of the pseudo-relevance feedback we were using earlier in Run 6). This run was to

simulate "long-term relevance feedback". The rationale was that if one user submitted a query and clicked on certain relevant results, then we could collect the relevance judgments and learn from them to generate a better ranking for the same query in the future when a different user submits the same query (or a similar query).

9. **1276859235707__PhybaselineRelfbWMD_25_0.2sub:** This run is similar Run 8 but the relevance feedback was applied on top of Run 7 instead of Run 6. Thus it was also to simulate "long-term relevance feedback". The main difference between Runs 8 and 9 is that Run 8 used top-N-based MeSH feedback whereas Run 9 used distribution-based MeSH feedback. Thus, comparing Run 8 and Run 9 can allow us to further compare the two MeSH feedback methods in the setting of relevance feedback.
10. **1276862759027__PhybaselineRelFbWMR_10_0.2_top20sub:** Similar to Run 8, but this time we use the relevance judgments provided from the top 20 documents to re-rank only the *remaining* documents. This was to simulate "short-term relevance feedback" (within the same search session). The rationale was that once we know a user found certain results as relevant, we can then simply use these judgments to generate a better ranking of the remaining (unseen) documents (i.e., documents originally ranked below top 20). In this case, the same user who provided feedback (i.e., relevance judgments) can benefit from his/her own judgments.

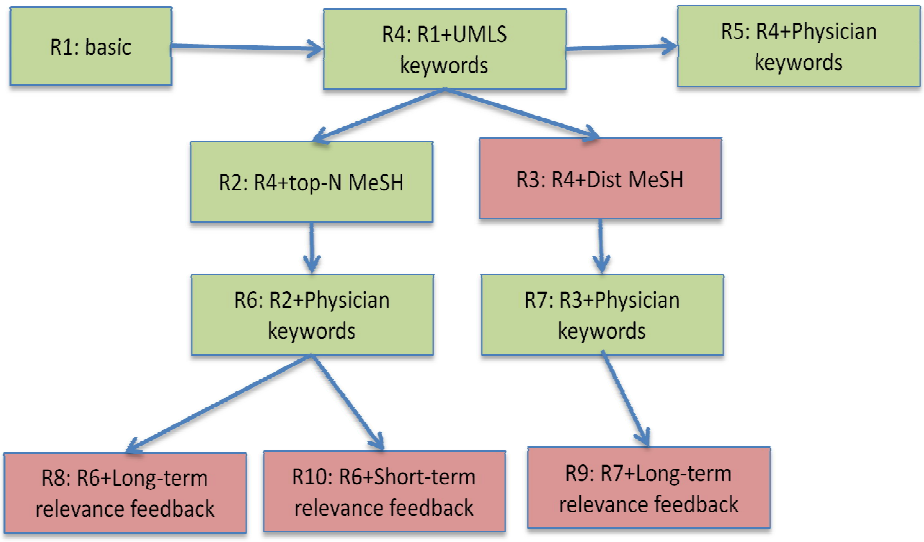

Fig. 2. Dependency of different runs. Green boxes are the runs that improve MAP score from their parent runs. Red boxes are the runs that reduce MAP score from their parent runs.

Table 1. Summary of results for different runs

Run ID	Type	Retrieval Methods	MAP for all	Prec@10	Recall
1	Automatic	Baseline Method (Standard Retrieval Model)	0.2754	0.4286	0.8081
2	Automatic	Run 4 + topN-Mesh-FB	0.2902	0.4429	0.8464
3	Automatic	Run 4 + distr-Mesh-FB	0.2626	0.4	0.8464
4	Automatic	Run 1 + KeywordWeighting	0.2808	0.4429	0.8464
5	Manual	Run 4+ PhysicianKeywords	0.3441	0.4714	0.8618
6	Manual	Run 2+ PhysicianKeywords	0.3551	0.4714	0.8618
7	Manual	Run 3+PhysicianKeywords	0.3441	0.4714	0.8618
8	Manual	Run 6 + LongTermRelFB	0.3059	0.4571	0.8292
9	Manual	Run 7 + LongTermRelFB	0.2837	0.4571	0.8292
10	Manual	Run 6 + ShortTermRelFB	0.2713	0.4286	0.8292

The performance figures of our 10 runs are shown in Table 1. We observe that in terms of MAP, apart from Run 3 and Run 10, all other runs outperform the baseline run 1, indicating most of our extensions of the standard retrieval model are effective. Also, the manual runs usually perform better than the automatic runs, suggesting that the additional information obtained from the physicians is beneficial.

Looking at specific run comparisons, we can further draw the following conclusions:

1. **UMLS-based keyword reweighting:** Given that the performance of Run 4 based on all three metrics is better than Run 1, we can conclude that UMLS based keyword weighing is an effective strategy for improving retrieval performance for this task.
2. **Top-N-based vs. distribution-based MeSH feedback**: Given that the performance of Run 2 is better than Run 4, we can conclude that the top-N-based MeSH feedback method is effective and it can be added on top of UMLS-based keyword weighing to further improve performance. However, Run 3 is worse than Run 4, indicating that the distribution-based MeSH feedback method did not work well. These results, along with other results where we can compare these two feedback methods (i.e., Run 6 vs. Run 7; Run 8 vs. Run 9), all suggest that the top-N-based MeSH feedback method is more effective than the distribution-based feedback method. We suspect that the poor performance of distribution-based feedback may be due to the suboptimal setting of parameter values. We intend to further explore this issue in the future to better understand the cause.
3. **Physician keywords:** The runs 5, 6 and 7 perform considerably better than their corresponding baseline runs, i.e., Runs 4, 2, and 3. We can thus conclude that the additional keywords provided by physicians were greatly helpful. This result also provides us with some insight on how the case queries should be formulated. Apart from taking the actual case description as an input (something that was already available in the query), we can ask the users to provide a separate set of related

keywords that they feel may be present in the relevant document. These can be assigned low weights (to avoid "query drift") and then used for query expansion. In other words, physicians can potentially formulate a more effective query than the case description in a current query.

4. **Additive benefit of different heuristics:** The continuous improvement in performance from run 1 to 4 to 2 to 6 shows that our strategies at dealing with different challenges continue to work well when combined. Specifically, UMLS keyword reweighting, top-N-based MeSH feedback, and physician keywords can be combined to achieve additive benefit, leading to the best performing Run 6.
5. **Relevance feedback:** The performance of relevance feedback based runs, 8 through 10, is lower compared to the corresponding baseline runs 6 and 7. This is a somewhat surprising result since in relevance feedback, we have available relevance judgments from the users for top-20 results which is a significant advantage over other runs. We thus expected that our relevance feedback runs would outperform all other runs. Our analysis suggests that in certain cases, the documents judged by our physician users as relevant were not judged as relevant in the official gold standard, and as a result, these relevance feedback runs have over-fitted the user-selected relevant documents, leading to inferior performance. The observation hints at a certain level of subjectivity being involved in the case retrieval problem. Due to the limited time available for preparing this paper, we did not have time to further analyze the reason, but this issue clearly warrants further experimentation, and we plan to explore it in the future.

We also did a comparison of our submitted runs with all the submissions from other participants of this task in terms of the Mean Average Precision (MAP) on all the topics and found that our 10 runs were ranked 1 through 7 and 9 through 11 among all the submissions. Additionally in 9 of the 14 topics, one of our runs performed the best among all the submissions. Figure 3 shows the MAP scores of different teams. Runs 16 through 25, highlighted in red are ours while the remaining shown in blue are from other teams. One interesting observation is that our baseline run (i.e., Run 1, or 1276844704028__baselinefbsub) actually performed very well as compared with other submissions; indeed, only one run in the pool of submissions from other groups is slightly better than this run in MAP. Since this run represents a well-tuned standard retrieval model, this result suggests that such a well-tuned standard language modeling approach to retrieval remains a strong competitor for this task. Our overall superior performance has also clearly benefited from the use of this strong baseline method. However, it is very encouraging to see that several heuristic extensions that we developed can further outperform this strong baseline method, suggesting that there is clearly potential to further improve a state of the art general retrieval method for this special retrieval task by leveraging domain knowledge and user feedback.

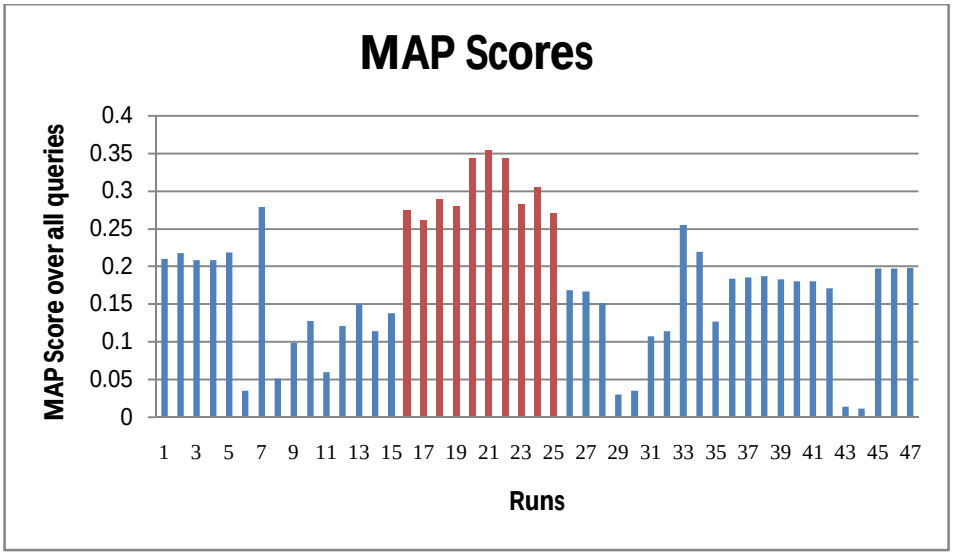

Fig. 3. MAP scores over all queries for all submitted runs. Our runs are highlighted in red

7 Conclusions

In this paper we described the details of our participation in the ImageClef 2010 medical case retrieval task. Our focus was primarily at identifying the major challenges arising from the differences between general retrieval and medical case retrieval, and then at developing methods for addressing them. We observed that taking into account the semantics of query keywords helped in assigning more appropriate keyword weights, and proposed a UMLS-based keyword reweighting strategy which is shown to be effective. We also proposed two novel methods (i.e., top-N-based and distribution-based) for leveraging MeSH terms to perform pseudo feedback and automatically re-rank documents. The results show that the top-N-based method is more effective than the distribution-based method, and it can be combined with other heuristics to further improve retrieval accuracy. Finally we explored ways of dealing with the vocabulary gap issue, and found that additional related keywords provided by physician users can be used with low weights along with the original query case to greatly improve both precision and recall. However, while we expected relevance feedback to be beneficial, our results show that all the feedback runs were worse than their corresponding baseline runs, an issue to be further looked into. Overall, most of our strategies helped improve performance and our methods largely outperformed those other participating groups.

Due to the time limit, we have not yet been able to conduct a thorough analysis of our experiment results and all the proposed methods. In the future, we will run additional experiments and perform more analysis of the proposed methods, in particular to better understand why relevance feedback did not help.

Acknowledgments. This paper is based upon work supported in part by an IBM Faculty Award, an Alfred P. Sloan Research Fellowship, and by the National Science Foundation under grants IIS-0713581 and CNS-0834709.

References

[1] Müller, H., Kalpathy-Cramer, J., Eggel, I., Bedrick, S., Kahn, C.E., Hersh, W.: Overview of the CLEF 2010 medical image retrieval track. In the *Working Notes of CLEF 2010*, Padova, Italy, 2010.

[2] Simpson, M., Rahman, M.M., Demner-Fushman, D., Antani, S., Thoma, G.R.: Text- and Content-based Approaches to Image Retrieval for the ImageCLEF 2009 Medical Retrieval Track, In the *Working Notes of CLEF 2009*. URL: http://www.clef-campaign.org/2009/working_notes/CLEF2009WN-Contents.html

[3] Zhai, C.: *Statistical Language Models for Information Retrieval (Synthesis Lectures Series on Human Language Technologies)*, Morgan & Claypool Publishers, 2008

[4] Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval , *Proceedings of the Tenth ACM International Conference on Information and Knowledge Management* (CIKM'01), pages 403-410, 2001