

# Facilitating Biomedical Systematic Reviews Using Ranked Text Retrieval and Classification

David Martinez    Sarvnaz Karimi    Lawrence Cavedon    Timothy Baldwin

NICTA Victoria Research Laboratory  
The University of Melbourne  
Victoria 3010, Australia

{davidm,skarimi,lcavedon,tim}@csse.unimelb.edu.au

**Abstract** *Searching and selecting articles to be included in systematic reviews is a real challenge for healthcare agencies responsible for publishing these reviews. The current practice of manually reviewing all papers returned by complex hand-crafted boolean queries is human labour-intensive and difficult to maintain. We demonstrate a two-stage searching system that takes advantage of ranked queries and support-vector machine text classification to assist in the retrieval of relevant articles, and to restrict results to higher-quality documents. Our proposed approach shows significant work saved in the systematic review process over a baseline of a keyword-based retrieval system.*

**Keywords** Information Retrieval, Machine Learning.

## 1 Introduction

The growth and applicability of evidence-based medicine (EBM) has enormous potential for the way medical treatments are applied throughout the world. However, the task of preparing systematic clinical reviews for EBM is currently human workload-intensive. A systematic review is generally formulated against a specific clinical question, such as: *In a pre-hospital setting, what is the effect of intubation vs no intubation in traumatic brain injury?*

As a general framework, systematic reviews are conducted using the following main steps [3]:

1. formulate a high priority problem and develop the *inclusion* criteria (i.e. criteria for judging an article as relevant to the clinical question);
2. *search* through all the relevant published studies or articles. This step involves formulating a complex boolean query and submitting it to a number of databases of medical literature;
3. assess eligibility of retrieved articles, and extract data. The *assessment* involves judgement against the inclusion (and possibly exclusion) criteria;
4. analyse and present findings;
5. interpret results and draw conclusions.

Proceedings of the 13th Australasian Document Computing Symposium, Hobart, Australia, 8 December 2008.  
Copyright for this article remains with the authors.

While almost all these steps are labour-intensive, the second and third steps are particularly important, yet time-consuming. General practice in performing this *searching* stage involves developing boolean queries over well-known medical databases, such as Ovid MEDLINE, PubMed, or EMBase. Forming these queries, often as long as sixty lines, is not straightforward, with search experts constantly modifying their queries and seeking for keywords to be augmented to the query to improve recall. Also, boolean queries, even if potentially effective in retrieving most of existing relevant documents, do not rank the retrieved documents, and therefore all the returned articles must be scanned to specify candidates to be studied in detail.

Further, the *assessment* stage is often formulated in two steps:

1. reading the abstracts returned by the boolean query to determine if the associated papers are candidates for the review. This list may number in the tens of thousands;
2. retrieving and reading full-text documents for those which are included, and judging further which of these should be included in the review itself. This collection may number in the hundreds or thousands.

In this paper we investigate this searching problem in the framework of keyword-based text information retrieval (IR) and text classification. We are particularly interested in fully or partially migrating the searching step from boolean to ranked, as implemented in most popular search engines. Differences between retrieval for systematic reviews and standard IR tasks makes this problem challenging. In the searching stage of a systematic review, there is no simple specific topic to be looked up. Queries can be a combination of subject of the review, research questions to be addressed, and inclusion criteria to be observed. Unlike standard IR, recall is of crucial importance to ensure that no important evidence in regard to a clinical question is overlooked.

We investigate a two-stage search system that initiates a search using initial information on a priority research topic, then through a re-ranking scheme based

on text classification assist users to find relevant information more quickly. In particular, we estimate the potential amount of work saved by using such an automated system, as compared to performing manual reading and checking of all results returned by a boolean query, as is current practice.

## 2 Background

A number of organisations, such as the Cochrane Collaboration<sup>1</sup> and the Agency for Healthcare Research and Quality (AHRQ)<sup>2</sup> publish systematic reviews, as well as associated data for each step in the process: the boolean queries, search results, and inclusion criteria used. The enterprise of producing reviews, as well as updating the existing ones, can be massively time-consuming: a systematic review for a single clinical question may take a number of person-years to compile. Hence, any automated support for the process has the potential to be extremely valuable.

Work has recently been performed in improving the search process so that a higher quality set of documents are retrieved by the first of the search steps. In particular, document classification techniques are used to filter the set returned by the search, and thereby reduce the work of manually reviewing the articles that are candidates for inclusion in the final review. Further, the presence of audit data for existing reviews provides a valuable resource for evaluating performance and effectiveness of the techniques developed.

Cohen *et al* [2] specifically address the issue of reducing workload involved in preparing systematic reviews on specific classes of drugs. They construct a classification system, using a voting perceptron classifier [5], trained on data associated with 15 drug reviews published by AHRQ. The document set for each review topic was the set of MEDLINE abstracts returned by the initial search associated with that review, limited to those which were also contained in the TREC 2004 Genomics Track document corpus (so that full-text papers would be available). The classifier's feature set was constructed from these abstracts. Features included: bags-of-words constructed from title and abstract; MeSH (Medical Subject Headings) terms associated with the abstracts; and MEDLINE publication type. Inclusion and exclusion results for each abstract, as published by AHRQ, were used to create the classification gold standard.

Cohen *et al* evaluate their classifier using  $5 \times 2$  cross-validation on the document set. They report recall and precision for each of the 15 drug reviews, as well as a measure of *work saved*, which is designed to more closely reflect the actual effectiveness of the classifier in the context of the task. *Work saved* is the percentage of papers that meet the published inclusion criteria which would not have to be manually inspected

(because they were filtered out by the classifier). In particular, Cohen *et al* report work saved over random sampling at recall of 95% (wss@95). This metric is described in greater detail in 4.1.

The actual effectiveness of their technique varied with topic. For 11 of the 15 review topics, wss@95 was above 10%, which Cohen *et al* considered to be a minimum threshold of the technique adding value; it was estimated that this level would actually result in a saving of a person-week of effort. Three of the review topics resulted in a saving of over 50%.

Other research that uses text classification techniques in the context of EBM do not attempt to directly estimate “work saved”. Aphinyanaphongs *et al* (ATSHA) [1] apply a number of techniques — Naïve Bayes, AdaBoost, and Support Vector Machines — to classify documents in various content areas: etiology, prognosis, diagnosis, and treatment. Their evaluation involved comparison with a baseline technique, developed by Haynes *et al* [6], which uses PubMed clinical queries to the above four areas. The EBM source for both baseline and ATSHA's system was the ACP Journal Club<sup>3</sup>. The ACP Journal Club has expert clinicians who categorise articles from a broad set of journals into categories including the ones listed above: this categorisation was the gold standard. ATSHA created filters from a collection of MEDLINE records corresponding to 49 journals referenced by the ACP Journal Club over a given time period, and these were used to filter articles from those journals into the 4 categories. ATSHA found that their machine-learning based categorisation generally outperformed the query-based categorisation<sup>4</sup>, and argue that there is a significant reduction in workload over both the manual review method and over the time required to develop the query-filters.

MScanner [7] is a recent, more general-purpose biomedical classifier used to filter search results; it was designed for database creation/curation, rather than creating EBM reviews, with an emphasis on speed. MScanner uses a Naïve Bayes classifier and a compact feature representation to support the processing of the whole MEDLINE collection in a reasonable time (approximately 90 seconds). Poulter *et al* [7] describe its effectiveness on a specific classification task as compared to the use of an expert-developed PubMed boolean query: on a task with 3,544 results (1,089 relevant, 2,465 irrelevant), MScanner was comparable to the hand-crafted query in recall and precision up until about 900 results.

## 3 A Two-Stage Ranking System

We propose a ranking system that pipelines a generic text retrieval search engine, and a classifier that re-ranks the retrieved documents as demonstrated below.

<sup>1</sup><http://www.cochrane.org/>

<sup>2</sup><http://www.ahrq.gov/>

<sup>3</sup><http://www.acpjc.org>

<sup>4</sup>A slight drop in performance was noted for the diagnosis category, attributed to the small number of positive training examples.

### 3.1 Text Retrieval for Systematic Reviews

As mentioned earlier, common practice in literature review of medical articles is centred around boolean retrieval. Boolean retrieval has two main disadvantages: first, there is no ranking available in its output list, and second, it requires search expertise to formulate effective complicated queries. To facilitate this process for systematic reviews some pre-defined prototype templates have been defined for insertion into boolean queries. For example, if consideration should be restricted to only a particular publication type, such as *randomised controlled trials*, then a pre-formed Ovid format boolean query such as that below can be used as part of the main query [6]:

*randomised controlled trial.mp OR  
randomised controlled trial.pt*

where *.mp* indicates the term should appear in the title, abstract, or MeSH headings<sup>5</sup>, and *.pt* indicates *publication type*. For capturing topical features of the review, however, search experts need to specify appropriate keywords, and their arrangement in the query. For example, if a review's focus is pre-hospital intervention, the query might include:

*prehospital.tw.  
pre-hospital.tw.  
paramedic\$.tw.  
ambulance\$.tw.  
out of hospital.tw.  
emergency rescue.tw.  
emergency resus\$.tw.  
emergency triage.tw.*

where *.tw.* indicates the search should be applied to text words of title or abstract. It is worth noting that stemming as is practised in IR systems is forced by explicitly using \$ in such queries and it is therefore more easily transferable to a ranked system than other features.

In contrast, a ranked retrieval system provides a ranked list, and querying is easier for inexperienced users. However, it does not provide specific features of a boolean system, such as recursive operators (e.g. nested AND and OR), or searching over specific metadata available in the MEDLINE records or other medical or clinical text collections.

A systematic review is built on a priority research question that is generally composed of four information components: prevention, intervention, comparison, and outcome; these are collectively known as PICO. *Prevention* specifies for which group of patients the information is targeted; *intervention* specifies the medical event whose effect is under investigation; *comparison* is the evidence of producing better or worse results

against other interventions or no intervention; and *outcome* specifies the effect of intervention. For instance, a question such as “*In the pre-hospital setting, what is the effect of intubation vs. no intubation in traumatic brain injury (TBI)?*” is composed of “traumatic brain injury” as prevention, “prehospital intubation” as intervention, “no intubation” as comparison. Outcome is not specifically listed in the review primary question but would be listed in its inclusion criteria.

Boolean queries normally cover one or two of these question components (mostly prevention and intervention), and the rest are inspected manually in the articles retrieved and marked to be further investigated (we refer to these as the *first tier* judgements). The remaining components are reported in the review as *inclusion criteria* or sometimes as *exclusion criteria* (listing articles that did not comply with some criteria). Examples of such inclusion criteria can be language — as only English languages studies be eligible — or publication date. Such criteria can be specific to the topic as well. For example, an inclusion criteria for the example review topic above could be “mortality” as outcome.

In addition to the main research question targeted in the review, there may be some research subquestion under the main review title. Some of these questions specify disjoint subsets of the final set of articles to be included in the review. They are sometimes covered by different boolean queries, which are reported separately in the review.

All of the above-mentioned characteristics of the search problem in systematic reviews make the query formulation process difficult. Many options require investigation for this domain, both for boolean and ranked queries. To make the process more effective and time efficient, specially if ranked queries are considered as a replacement for long boolean queries, studies on formulating good ranked queries need to be pursued.

In its first stage of ranking, our retrieval system uses a search engine to run selected ranked queries and generate a ranked list of retrieved articles. The main concern, however, is forming ranked queries. Candidate information sources for developing such queries include: review title; PICO; detailed research questions; and inclusion criteria. In our experiments we explore formulating ranked queries and evaluate them in terms of retrieval effectiveness.

### 3.2 Re-ranking via Text Classification

The complexity of the task explained in the previous section implies a need for either a powerful ranked retrieval procedure that captures all the inclusion criteria in the review; or, if a traditional ranking is used, it must be complemented with other components to help to retrieve as many relevant documents as possible. We are interested in a system which first and foremost has high recall, and as a secondary desideratum, retrieves eligible papers precisely. It also must reduce the workload in systematic reviewing as much as possible. There-

<sup>5</sup>MeSH are the Medical Subject Headings, an taxonomy of medical terms which are manually ascribed to every entry in MEDLINE

fore, we study the effect of document classification in a ranked system framework.

We rely on support vector regression (SVR) [4] to re-rank the output of the text retrieval system. The basic idea of regression algorithms is to find a function that approximates the training instances by minimising the prediction error. The main difference between SVR and other regression methods is that a user-specified parameter  $\epsilon$  defines the lower limit from where deviations are considered. SVR also tries to maximise the “flatness” of the function at the same time as minimising the error, that is, it tries to fit all training instances within a margin of width  $2 \times \epsilon$ . There is also a tradeoff with the prediction error, since it may be necessary to allow some training instances to have nonzero error in order to build a better function. This is controlled by the parameter  $C$ .

For our experiments, we used the Weka machine learning toolkit [8] with first order polynomial kernels and default parameters ( $\epsilon = 0.001$ ,  $C = 1$ ). In order to train our classifier we took the top-ranked documents (from the retrieval engine) at different cutoffs. The trained model was then applied to the rest of the collection, and finally the documents were ranked according to their weight. Different sizes of training data were tested to better analyse the classifier’s effectiveness: the top-ranked 10, 20, or 30 percent of the documents were used.

As our feature representation we chose two feature sets. Our basic feature representation consists of a bag-of-words model including all words occurring in the “abstract” and “references” sections of the paper. For our second set of features, we extended the first set with the MeSH headings of the articles. We performed separate experiments in order to measure the impact of the hand-annotated MeSH terms in the results.

## 4 Experimental Data

MEDLINE is the most popular database of articles of medical articles freely available to the research community. A recent collection of 16,676,340 abstracts was used in this study as document collection. For every article indexed by MEDLINE, there is an entry — known as a *citation* — which contains the title and abstract of the article, accompanied by metadata. The metadata includes publication date, language, author information, MeSH headings associated with the article at the time of its publication, publication type and venue, and a unique identifier known as a PMID.

We selected 17 systematic reviews from the AHRQ collection (see Section 2) to test our system and form our queries. The selection process was based on the clear provision of included and excluded papers and search strategies; this means that relevance judgements were available for the queries (i.e. the documents returned by the boolean query). A list of included and excluded MEDLINE citations as indicated in the reviews were extracted using their provided PMIDs as the first level of relevance judgements (first-tier). We excluded

review	AHRQ identifier, Year	first-tier	second-tier
1	1, 1999	822	184
2	66, 2002	267	67
3	106, 2004	38	12
4	118, 2005	273	92
5	130, 2006	421	198
6	131, 2006	413	83
7	136, 2006	158	117
8	145, 2006	508	104
9	146, 2007	130	34
10	167, 2008	1,103	440
11	138, 2006	796	65
12	57, 2003	647	228
13	11, 1999	329	21
14	100, 2004	535	121
15	103, 2004	932	203
16	110, 2004	2,329	365
17	116, 2005	158	77

Table 1: Specifications of the selected reviews. The third column represents the total number of articles initially considered to be further investigated in details (first-tier), and the fourth column specifies the number of documents chosen to be included in the review (second-tier).

any listed paper that was not indexed in MEDLINE. The same process generated another set of relevance judgements on documents included in the final review (second-tier). Specifications of these reviews are listed in Table 1.

### 4.1 Evaluation

We evaluated our system on two levels: retrieval that generate the initial results from ranked querying, and the final re-ranked list. For the initial text retrieval evaluation, we use standard IR metrics: precision and recall.

For document classification, we calculate the WSS measure (work saved over sampling) as defined by Cohen et al [2]. WSS measures the number of documents in the collection that the user would need to hand-check in order to reach a fixed recall over the relevant documents. For example, if the fixed recall is 95% and the documents are randomly ordered, an average of 95% of the articles would have to be inspected to reach 95% recall. WSS measures the difference between a given ranking and a random sampling. Its formula is shown below

$$WSS = ((TN + FN)/N) - 1 + (TP/(TP + FN)), \quad (1)$$

where  $TP$ ,  $TN$ ,  $FN$ , and  $N$  represent the number of true positives, true negatives, false negatives, and total number of instances, respectively.

In this paper, we measure WSS at different points in the document ranking, considering the TOP-K as positive and the remaining as negative. For our main results we fix the recall to 95%, in line with the study by Cohen et al. [2]. To calculate the WSS metric, we compute recall for different K values until 95% recall is achieved. At this point, WSS@95 can be computed as shown in Equation 2

$$WSS@95 = ((TN + FN)/N) - 0.05. \quad (2)$$

Note that the number of relevant documents in the ranking ( $R$ ) will affect the minimum and maximum values of the metric. For  $WSS@95$ , since we have a fixed recall of 95%,  $FN$  will always be  $0.05 \times R$ ;  $N$  will be constant; and  $TN$  will determine the value of the ranking in saving work. Low  $TN/N$  ratios will produce negative  $WSS$  scores, since this indicates that we need to go to the bottom of the list to find all relevant documents, which would be outperformed by a random sampling.

It is worth explaining that ideally we would want to *a priori* approximate the  $K$  value that separates positive and negative instances in the classifier’s output. This can be a difficult problem, since we cannot be sure of the total number of relevant documents for a given query. One way of dealing with this issue would be to estimate the expected relevant documents by using training or held-out data as a reference.

Apart from  $WSS$ , we also calculated the *receiver operating characteristic* (ROC) curves and corresponding *area under curve* (AUC) values for the produced rankings. ROC curves illustrate the trade-off between the true positive rate and the false positive rate, providing a visualisation of the classifier’s performance at different cutoff points. The AUC score summarises the curve and the performance of the classifier in a single number, for easier comparison.

For the final results—since the text classification module also requires us to use the top documents for training, for a uniform evaluation over different training splits—we measured the  $WSS@95$  and AUC scores over the full collection, by putting the training documents at the top, and re-ranking the remaining according to the scores of the classifier. The reason for this is that the benefits from re-arranging the training data should not be credited to the classifiers.

## 5 Experimental Results

We first measure the performance of the existing boolean queries over MEDLINE. We then evaluate different retrieval strategies that rely on different sources of query-words. For our final experiment we rely on a text classification algorithm for improving the initial ranked results.

### 5.1 Boolean Retrieval

Each systematic review contains search strategies which list all the finalised boolean queries – for different databases – used to retrieve potential relevant articles. We extracted Ovid MEDLINE boolean queries from our selected 17 reviews and re-ran them against Ovid MEDLINE that indexes articles from 1950 to the first week of October 2008. We then extracted PMIDs of the retrieved articles and matched them against the relevance judgements we created from the reviews as reported in Table 1. Surprisingly however not all these

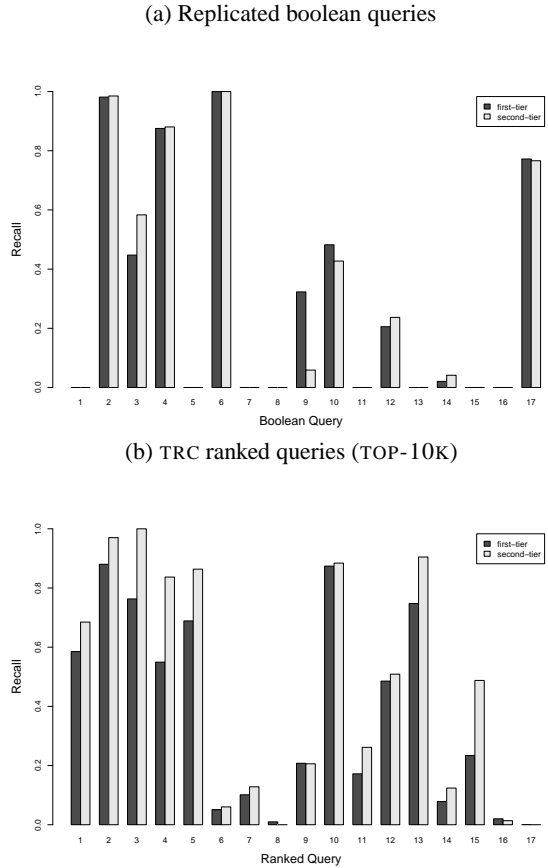


Figure 1: Recall of replicated boolean queries and ranked queries based on the relevance judgements reported in the reviews. Note 7 boolean queries (1, 5, 7, 11, 13, 15, 16) are just shown for easier comparison of the two graphs and they do not represent any data.

queries were in the state of reproducibility and led to errors when running them in Ovid. Examples of these errors were missing query lines in the reported queries, non-matching MeSH headings, and query lines referring to specific partial results in the query that made it dependant on the time it has been first executed. Figure 1 (a) shows recall values of 10 out of 17 queries only, as the remaining were not replicable. As presented in the figure, in contrast to our expectation, not all the relevant documents were retrieved by these queries, with some of them showing very low recall (query 8 and query 14 had 0.0 and 0.02 recall, respectively, based on the first-tier judgements). This clearly illustrates the limitations of the present approach for documenting systematic review strategies based on boolean queries.

### 5.2 Ranked Retrieval

Ranked retrieval is the first stage of our architecture, as explained in Section 3.1, which retrieves an initial set of documents to feed to the second stage, classification.

We created ranked queries from the 17 systematic reviews available from AHRQ based on three main in-

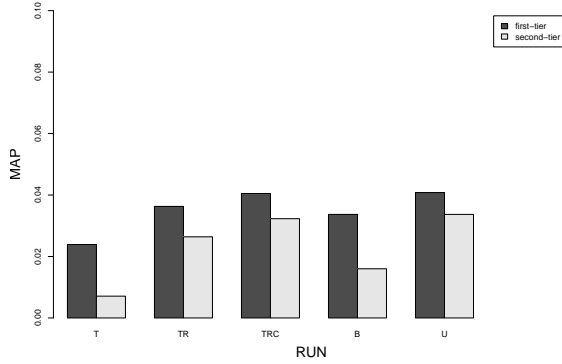


Figure 2: Retrieval effectiveness of different types of ranked queries on MEDLINE citations (TOP-10K).

formation components they are built on: title or major research question (T), other research questions (R), and inclusion criteria (C). For each of these query types, we used the exact string that appeared in the corresponding review. We also flattened boolean queries to make ranked queries by removing metadata indicators such as *.tw.* or *.pt.* (B). Finally, we ran some combinations of these queries against the MEDLINE collection. In our experiments, we used Zettair<sup>6</sup> search engine with its default setting for Okapi BM25 ranking scheme.

Figure 2 shows the mean average precision (MAP) for the five categories of title only (T), title and research questions (TR), title and research questions and inclusion criteria (TRC), ranked queries made from boolean queries (B), and using unique words of C with T and R (U). All categories of the queries work poorly, with MAP scores less than 0.1 using each of the first-tier and second-tier judgements. The third group, TRC, achieved a slightly better MAP score of 0.0405 in comparison to others, and we will consider this method as the baseline for the classification experiments. We also show recall values for each query in Figure 1 (b) as a comparison to their boolean counterparts where available. Interestingly, ranked queries are more effective based on the second-tier judgements, where only included articles are considered as relevant (twelve out of seventeen queries had higher recall in the second-trier level than the first-trier level). Also, comparing the recall values of the boolean and ranked queries in the second-trier level, five ranked queries had higher recall than boolean queries, with boolean queries winning only for four queries.

Recall values specify a maximum threshold on our system effectiveness in finding relevant documents. That is, our classifier can improve the ranking of these documents in the list, but no new relevant document that might have been missed in the initial retrieval can be added to the pool.

Systematic reviews are expected to cover all the relevant studies and therefore recall is crucial. In order

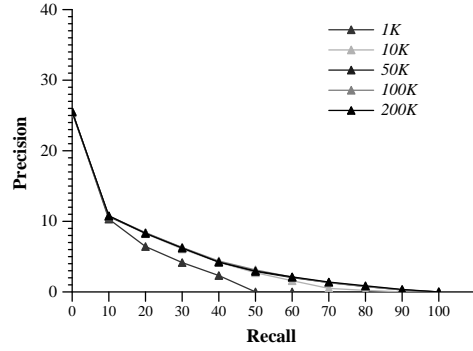


Figure 3: Precision at eleven recall points for different cutoffs when using queries composed of title, research questions, and inclusion criteria (TRC). Judgements were first-tier articles.

to choose an experimentally sound cutoff point in our ranking list that covers most relevant documents, we calculated precision at eleven recall points for the different cutoffs of TOP-1K, TOP-10K, TOP-50K, and TOP-100K, averaged over 17 queries (Figure 3). There was little difference between recall values when more than ten thousand documents were retrieved. We therefore chose TOP-10K results as input to the classifier. This number of documents also reflects the amount of articles that researchers commonly manually check after running boolean queries, which can be in the order of tens of thousands.

### 5.3 Re-ranking via Text Classification

The goal of the re-ranking step is to improve the retrieval rankings by moving relevant documents towards the top of the list. As mentioned above, achieving very high recall is crucial, and the value of a ranked list will be dependant on the number of documents required to reach a given recall (95% in our case), which can be shown by the metrics WSS@95 and AUC (see Section 4.1).

For our classification experiments we rely on the top 10,000 documents from the text retrieval step, using the TRC query strategy, which attained the highest recall. As relevance judgements, we focused on the documents included in the final reviews (second-tier); the main reasons being that: (i) they are the ones chosen for the final review, and (ii) the retrieval step showed that the recall is proportionally higher for these.

Table 2 shows the WSS results and TOP-K value for our first experiment relying on the basic feature set (no MeSH terms), with different amounts of training data. The TRC column shows WSS@95 without re-ranking; the average WSS score for these is 16.4%, which would be the amount of work saved over random sampling of the documents. The classifier is able to clearly improve these values for different amounts of training data, obtaining the best performance for all but three of the queries. The results using 10% of data are significantly improved, according to the paired t-test, and we save

<sup>6</sup><http://www.seg.rmit.edu.au/zettair/>

Query	Baseline		Classifier					
			10%		20%		30%	
	wss	K	wss	K	wss	K	wss	K
1	8.9	8608	19.4	7564	9.0	8599	16.9	7812
2	19.4	7562	53.3	4169	44.7	5030	38.4	5660
3	11.0	8398	56.5	3847	56.7	3830	55.8	3925
4	17.9	7710	67.9	2707	41.5	5348	42.4	5259
5	45.1	4986	67.7	2728	67.6	2742	59.4	3558
6	42.1	5288	42.1	5288	42.1	5288	63.2	3179
7	8.7	8631	69.6	2537	54.9	4010	59.8	3517
8	-	-	-	-	-	-	-	-
9	59.0	3597	33.0	6195	18.4	7655	13.6	8135
10	1.6	9344	26.6	6835	33.4	6163	31.2	6375
11	2.7	9230	2.2	9281	25.2	6977	-3.0	9799
12	13.4	8165	7.6	8736	5.8	8916	5.0	9003
13	8.7	8628	5.4	8958	0.7	9426	-2.3	9729
14	2.5	9247	-2.3	9731	24.0	7096	16.0	7896
15	4.5	9048	7.7	8728	4.5	9054	2.0	9296
16	-0.1	9506	-0.1	9506	13.7	8133	63.5	3146
17	-	-	-	-	-	-	-	-
Avg.	16.4	7863.2	30.5 <sup>†</sup>	6454.0	29.5 <sup>†</sup>	6551.1	30.8	6419.3
Wins	3		6		4		2	

Table 2: Comparison of the baseline (TRC) and the two-stage system with classifier using WSS metric with basic features. K: number of documents that are returned as positive. Wins: number of queries for which a particular approach attains the best results. †: paired t-test with 95 percent confidence level (over baseline).

more than 30% of the work on average. Notice also that for queries 8 and 17 there are no results, since the TRC query-strategy is not able to return any relevant document.

As the training data gets bigger we can see that the WSS score remains very similar on average. The main reason for this is that even if the classifier should get more accurate with additional training data, the top of the ranking will be given by TRC, and the pool to re-rank will be smaller, forcing the classifier to do a better job to obtain the same WSS score. We can see that the results for different training splits change depending on the query, but the averages remain the same. The paired t-test shows that there are differences between 10% and 20%, but not between 20% and 30%. These figures illustrate that 10% of training data is enough to benefit from the text classification system.

We also calculated the more known ROC curves and corresponding AUC values for this experiment. The results are given in Table 3, and they show a similar behaviour, with the 10% classifiers obtaining slightly better curves than other training splits. In this case the differences are always significant according to the t-test.

Our next step was to add MeSH terms as features. A summary of the WSS@95 and AUC results is given in Table 4, with the baseline feature results for reference. We can see that MeSH terms improves the results when using 20% of the data for training. This indicates that the classifier improves considerably in ranking the remaining 80% of data.

The WSS results for each query are shown in Figure 4, when using MeSH terms. Here, we can see that there are big differences depending on the query, with cases where the classifiers make a huge contribution

Query	Baseline	Classifier		
		10%	20%	30%
1	69.6	75.9	73.4	74.6
2	85.2	89.7	89.6	88.2
3	89.8	92.3	93.6	93.5
4	86.7	91.2	88.9	87.9
5	90.6	93.1	92.4	91.5
6	66.4	66.4	66.4	74.7
7	68.9	90.8	83.4	79.8
8	-	-	-	-
9	86.6	83.7	77.7	74.4
10	61.3	79.4	75.9	71.3
11	56.2	75.3	74.1	66.3
12	66.3	72.1	72.3	69.6
13	84.9	85.7	86.5	84.9
14	68.2	71.6	80.7	72.6
15	71.2	75.3	74.8	74.3
16	57.5	57.5	67.7	76.1
17	-	-	-	-
Avg.	73.9	80.0 <sup>‡</sup>	79.8 <sup>‡</sup>	78.6 <sup>†</sup>

Table 3: Comparison of the baseline (TRC) and two-stage system that uses classifier based on AUC metric. †, ‡: paired t-test with 95 and 98 percent confidence levels respectively (over baseline).

Metric	Baseline	Feats.	Classifier		
			10%	20%	30%
WSS	16.4	Base	30.5	29.5	30.8
		MeSH	31.5	34.3 <sup>†</sup>	31.2
AUC	73.9	Base	80.0	79.8	78.6
		MeSH	80.5 <sup>‡</sup>	80.6 <sup>‡</sup>	79.0 <sup>‡</sup>

Table 4: Comparison of average WSS and AUC between Base features and adding MeSH headings to the Base ones. †, ‡: paired t-test with 95 and 98 percent confidence levels respectively (MeSH over baseline).

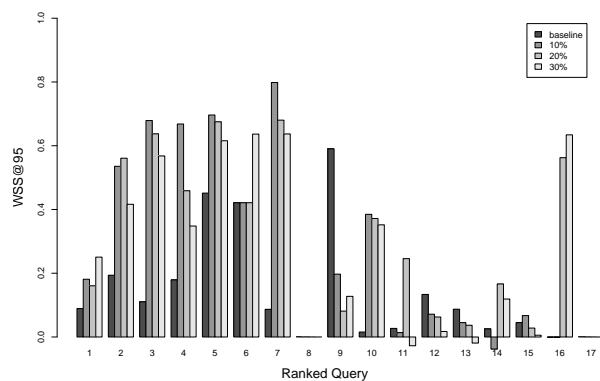


Figure 4: Word saved over sampling (wss@95) per query when MeSH terms are used along with text of articles as classifier training features.

(e.g. queries 7 and 16), and others where the scores go down (e.g. queries 9, 12 and 13). The results for query 9 are particularly interesting; this is the query that has the highest baseline, but the performance of the re-ranking system clearly drops. On the other hand, for query 16 the baseline ranking is as low as random sampling, but the classifier is able to obtain remarkable results. We think that there are a number of parameters affecting the final performance, such as the baseline score or the number of relevant documents in the collection, which would give us a better indication of the expected performance per query. Overall, we can see that the re-ranking approach is beneficial for most queries.

## 6 Conclusions and Future Work

We addressed the search needs for building systematic clinical reviews for EBM, an increasingly growing area that targets the way medical care is provided. This problem is specifically difficult to solve with standard search strategies. We illustrated some of the main problems of the current approach, which relies on boolean queries: they are time-consuming to formulate and maintain, they are difficult to execute without expert knowledge, and they do not provide a ranking of documents. Furthermore, when replicating existing search strategies from the AHRQ collection (publicly available on the web) we found that some are poorly documented, and those that can be replicated do not lead to the set of documents that was used in the construction of the final review.

Thus, we explored the use of ranked queries and text classification for better retrieval of the relevant documents. We found that different keyword-search strategies can reach recall that is comparable and sometimes better than the costly boolean queries. Use of ranked queries for systematic reviews was not explored before in the previous studies. In our next step we found that these retrieval rankings can be re-organised using machine learning to significantly

reduce the amount of work required to find most of the relevant documents. These results show good potential for the migration from boolean queries towards ranked systems, which are easier to maintain and provide means to prioritise the document analysis.

For future work our aim is to integrate our experimental findings into a new tool to aid in the construction of systematic reviews, focusing on the *search* and *assessment* steps. This tool would benefit from the user's feedback to dynamically re-rank the documents remaining to be analysed, and reduce the time to generate and maintain the reviews. Another important issue that we are exploring is the way to estimate the total number of documents to be checked to reach the required recall. We plan to address this issue by using similarity thresholds between the relevant documents already identified and the remaining candidates. Finally, we also want to enrich the features of our text classifier by adding different types of information, such as citation contexts.

**Acknowledgements** 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.

## References

- [1] Y. Aphinyanaphongs, I. Tsamardinos, A. Statnikov, D. Hardin and C. F. Aliferis. Text categorization models for high-quality article retrieval in internal medicine. *Journal of the American Medical Informatics Association*, Volume 12, Number 2, pages 207–216, 2005.
- [2] A. M. Cohen, W. R. Hersh, K. Peterson and P. Y. Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, Volume 13, Number 2, pages 206–219, 2006.
- [3] The Cochrane Collaboration. Cochrane handbook for systematic reviews of interventions, version 5.0.0, <http://www.cochrane.org/resources/handbook/>, 2008.
- [4] H. Drucker, C. J. Burges, L. Kaufman, A. Smola and V. Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, Volume 9, pages 155–161, 1997.
- [5] Y. Freund and R.E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, Volume 37, pages 277–296, 1999.
- [6] B. Haynes, K. A. McKibbin, N. L. Wilczynski, S. D. Walter and S. R. Werre. Optimal search strategies for retrieving scientifically strong studies of treatment from Medline: analytical survey. *British Medical Journal*, Volume 330, Number 7501, pages 1179–1182, 2005.
- [7] G. Poulter, D. L. Rubin, R. B. Altman and C. Seoighe. Mscanner: a classifier for retrieving medline citations. *BMC Bioinformatics*, Volume 9, Number 1, 2008.
- [8] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.