Extended Boolean Retrieval for Systematic Biomedical Reviews

Stefan Pohl **Justin Zobel** **Alistair Moffat**

NICTA Victoria Research Laboratory,
Department of Computer Science and Software Engineering
The University of Melbourne,
Victoria 3010, Australia
{spohl,jz,alistair}@csse.unimelb.edu.au

Abstract

Searching for relevant documents is a laborious task involved in preparing systematic reviews of biomedical literature. Currently, complex Boolean queries are iteratively developed, and then each document of the final query result is assessed for relevance. However, the result set sizes of these queries are hard to control, and in practice it is difficult to balance the competing desires to keep result sets to a manageable volume, and yet not exclude relevant documents from consideration.

Ranking overcomes these problems by allowing the user to choose the number of documents to be inspected. However, previous work did not show significant improvements over the Boolean approach when ranked keyword queries based on terms in the Boolean queries, review title, research question or inclusion criteria were used.

The extended Boolean retrieval model also provides ranked output, but existing complex Boolean queries can be directly used as formal description of the complex information needs occurring in this domain. In this paper we show that extended Boolean retrieval is able to find a larger quantity of relevant documents than previous approaches when comparable (or greater) numbers of documents are inspected for relevance.

Keywords: Information retrieval, extended Boolean retrieval, p-norm, effectiveness, systematic review, biomedical.

1 Introduction

Web search is one of the most prominent Information Retrieval (IR) applications. Typical question-answering scenarios are well supported by ranking highly the documents that not only look relevant by their content, but also receive external support such as by incoming links and anchor text references. In these applications, looking at one or a few of the highest ranked result documents might be sufficient, and if it is, the search process can be stopped. Commercial web search engines are optimized for this scenario and much IR research is focused on improving performance in the top, say 10, results.

However, if the objective is to carry out a comprehensive review for a particular topic, search cannot be stopped after finding a few relevant documents. In particular, reviews aim for very broad coverage of a topic, and seek to minimize any bias that might arise as a result of missed or excluded relevant literature. But the typical tensions in IR continue to apply, and if more relevant documents are to be found, more irrelevant documents will also need to be inspected. In the biomedical domain, systematic reviews of the whole corpus of published research literature (the largest collection, MEDLINE, currently indexes more than 17 million publications) are used to provide medical practitioners with advice to assist their case by case decision-making. To seed the reviews, complex Boolean queries are used on different citation databases to generate a set of documents which are then triaged by multiple assessors. In this domain, it becomes crucial to find as much of the relevant literature as possible for any given level of effort, because each item of overlooked evidence adds to the possibility of suboptimal outcomes in terms of patients' health-care.

The traditional Boolean retrieval model has been studied intensively in IR research. While it has straightforward semantics, it also has a number of disadvantages, most notably the strictly binary categorization of documents, and the consequent inability to control the result set size except by adding or removing query terms. For example, it is often the case that too many, or too few, or even no documents are returned, and no matter how the query terms are juggled, the "Goldilocks" point might be impossible to attain. In contrast, the broad adoption of ranking principles based on bag-of-word queries, and the resultant ability to order the set of documents according to a heuristic similarity score, means that for general IR applications users can consciously choose how many documents they are willing or able to inspect. Now the drawback is that bag-of-word keyword queries do not offer the same expressive power as Boolean queries do. Although extensions to the Boolean retrieval system have been suggested that produce a ranked output based on Boolean query specifications, they have not been broadly adopted for practical use – perhaps because, to date, simple keyword queries have typically been able to produce similar results, and, for lay users, are easier to generate.

Although ranking has the advantage of identifying a monotonically increasing total number of relevant documents as more documents are inspected, typical IR ranking functions face the difficulty that their ranking is dependent on properties of the whole collection, and can thus be difficult to reproduce, or even understand. Reproducibility helps in assessing review quality, and is thus often stipulated as a key requirement of comprehensive reviews [Sampson et al., 2008]. But if ranked queries are used, reproducibility can only be assured if all aspects of the computation are reported, including term weights and within-document term frequencies. With Boolean queries, all that is required is publication of the query that was used, together with the date or other identifying version numbers of the collections it was applied to. Moreover, previous work did not show improved retrieval results with ranked keyword queries compared to complex Boolean queries [Karimi et al., 2009].

In this paper, we show that, for the searching undertaken for the purposes of systematic reviews, an extended Boolean retrieval model finds more relevant documents

Figure 1: A typical query tree, showing different types of operators. Many of the details have been omitted.

than other current alternatives when typical numbers of documents are inspected, where typical is in terms of the result set size achieved by the initial complex Boolean queries specified by the researchers undertaking the review. Ranked results based on Boolean queries facilitate iterative Boolean query refinement through assessment of the top retrieved results and allows pure Boolean result sets to be extended if there is capacity to inspect more documents. As was noted already, finding more relevant documents with similar effort reduces the risk of missing important evidence that could affect decisions made by a medical practitioner.

2 Background

Evidence-based medicine aims to apply the latest published, scientific evidence in medical practice [Davidoff et al., 1995]. Systematic reviews of the current and prior literature are an essential tool to thoroughly and objectively survey the literature in order to address a specific research question. To minimize bias, the researcher's goal is to identify almost all of the publications reasonably related to the topic under review. Significant amounts of time are dedicated to the initial search phase, usually in the order of weeks or months [Zhang et al., 2006]. Search is performed using complex Boolean queries through interfaces such as `PubMed`[1] or `Ovid`[2], in multiple bibliographic databases like `MEDLINE` or `EMBASE`, each containing millions of citation entries. These entries usually contain titles and abstracts, and meta-data such as publication year, author(s), language, and manually indexed annotations in form of Medical Subject Headings (`MeSH`)[3]. After a final search strategy has been determined, all returned documents are triaged for their relevance based on their title, abstract and finally full-text. Other methods are also used to extend the set of relevant documents, such as following citations forward and backward, and hand-searching of conference proceedings. The evidence in each document in regard to the research question the review addresses is appraised, extracted, and finally synthesized [Higgins and Green, 2008]. To make the search reproducible, the search strategies used to locate the set of studies that are formally cited in the review are published along with the collated view of that literature that is provided by the review.

Queries are complex in multiple dimensions, notably operator richness, structure and size.

First, the operators used are beyond those in traditional pure Boolean systems. As well as binary conjunction (`AND`) and disjunction (`OR`) operators and their n-ary versions, field restrictions might be employed, or proximity operators, or wildcard term expansions, or `MeSH` terms and their so-called "explosion". All of these extensions must be supported within the syntax and semantics of the query language. For example, a query like

```
wom?n AND exp *Genomics/
```

matches all documents containing both a wildcard expansion of `wom?n`, such as `woman` and `women`, and with a main focus (the `MeSH *` qualifier) on `Genomics` or any `MeSH` heading below `Genomics` in the `MeSH` hierarchy, such as `Proteomics` (the `MeSH exp` operator).

Second, Boolean logic allows operators to be nested deeply, to express arbitrary concepts and the relations between them. Most queries follow a basic structure close to conjunctive normal form (CNF) that is generally referred to as faceted search in the literature [Hersh, 2008]. Semantically close terms, phrases, or `MeSH` headings are connected in disjunctions (using `OR`) which are then combined in a top-level conjunction (using `AND`). Often, further conjuncts are added that are based on meta-data, and act as filters corresponding to the inclusion or exclusion criteria of the review.

Finally, queries can become very long, and it is usual to split queries into different sub-parts and then subsequently combine the partial results by references to previous query line numbers. In total, queries typically consist of between a dozen and a hundred or more query lines, each containing terms, basic concepts, and simple operators; and

[1] `http://www.ncbi.nlm.nih.gov/pubmed/`
[2] `http://www.ovid.com/`
[3] `http://www.nlm.nih.gov/mesh/`

when written down as a single fully-expanded Boolean expression, can involve quite significant complexity.

Figure 1 shows parts of the query tree for one typical query. The range of operators that are used can be seen, and also the large number of nodes that are present as part of the query.

Because the queries are intended to be (re-)used later as filters against which newly published documents can be screened so that reviews can be periodically updated, they should generalize well and not over-fit to match on a possibly large fraction of documents retrospectively known to be relevant. In particular, while there is always a single Boolean query that returns exactly the set of all documents already known to be relevant, namely a disjunction of unique phrases drawn from each relevant document, this level (or any approximation of it) of over-fitting must be guarded against. Instead, the published queries are just the final statements used to generate the set of reviewed documents out of which a large fraction of the cited ones were drawn, and are thus formal descriptions of an information need, expressed in terms of Boolean logic.

2.1 Motivation

Boolean queries are best employed in data retrieval scenarios in which it is known beforehand what records exist, and which ones of them are to be retrieved. For example, searching through an email archive for a message with a particular document attached to it might proceed on the basis of knowing an approximate date of the email, the topic of the document, and the name of the sender of the email. With such cues available, Boolean fielded search can usually be relied on to locate the required item; and the searcher is satisfied immediately that a single email/document combination has been located.

Databases are a typical example of such controlled environments, and the manual indexing in bibliographic databases – and hierarchical classification schemes such as Dewey and Library of Congress – seek to add this feature to unstructured textual databases. But not all information needs can be foreseen, and providing a suitable set of indexing terms that covers all possible eventualities is tantamount to adding the whole of the document as index terms. This is why Boolean retrieval is also applied to unstructured textual data. However, the ambiguity of free-text, and the lack of a controlled vocabulary leads to the well-known precision-recall trade-off in IR, which notes that only a fraction of the documents in the returned set are actually relevant (the *precision* of that set of documents); and only a fraction of all relevant documents are in the returned set (the *recall* of that set of documents). These two competing requirements – high precision, so that the searcher's time spent examining documents is used to best effect; and high recall, so that the majority of the relevant documents are identified as part of the search – can be balanced by adjusting the query. For example, adding conjuncts to a query is likely to increase precision but decrease recall; and adding disjuncts is likely to increase recall but decrease precision. In the limit, if all documents are returned (by a disjunctive query containing terms of all documents), then recall is 1.0, but this is an unsatisfactory situation. Similarly, a query in which no documents are returned technically has precision of 1.0, but is equally useless (unless there are no relevant documents in the collection).

The Boolean retrieval model has a long history, but a number of main problems are repeatedly reported in the literature. For instance, novices and lay users may find Boolean queries difficult to formulate [Frants et al., 1999]. While this is an issue in general IR, a search intermediary, such as a librarian that knows the database, is usually part of a review team in this domain [McGowan and Sampson, 2005]. A bigger problem is that documents are only differentiated into two stark groups: those that match the query and are retrieved and inspected, and those that do not match and are not retrieved and thus never viewed, regardless of whether or not they are relevant. Moreover, in a typical conjunction-of-disjunctions query, documents are not retrieved if only one conjunct evaluates to false, and treated as if every conjunct had evaluated to false; and are retrieved regardless of whether all of the terms in each disjunct match, or just one. The documents in the retrieved set are then all treated identically and returned in either random order (database record number, for example), or according to some secondary sort criterion such as (reverse) date of publication. Since all documents in the final Boolean result set are inspected, it may seem that the presentation order is unimportant. But worth noting is that the creation of complex queries requires iterative refinement and assessment of query quality, and that this preliminary work is of necessity done on a subset of the documents returned by each trial query. In contrast to sampling the whole result set, probing the top results of a ranking based on relevance to the query could possibly reduce these costs. Finally, as has already been noted, the size of the result set of a Boolean query is largely out of the user's control.

Query Tree	# Docs	Fraction
AND	2,935	0.56
├── OR	258,560	0.67
│ ├── headache	37,758	0.22
│ ├── "muscle cramp"	1,617	0.11
│ └── ...		
├── OR	62,337	0.78
│ └── ...		
├── `human`	10,885,697	1.00
└── ...		

Table 1: Case study of a query with low overall success. The final column shows the fraction of the known relevant documents (based on all techniques, not just this one query) that are identified by that subexpression.

It is thus not surprising that analysis of the query performance in this domain shows that many of them actually do not find all documents finally included in reviews [Dickersin et al., 1994, Martinez et al., 2008]. Error analysis for low-success queries revealed situations such as depicted in Table 1. Although a reasonable fraction of the documents known to be relevant match with each of the conjuncts comprising the overall query, overall success is (naturally) not greater than that of the least successful conjunct. This is the price paid to reduce the result set size for the individual conjuncts to a reasonable size, and is typical of the patterns observed for Boolean queries. In particular, while not all of the relevant documents contain all the required query concepts, the query would in part have been constructed so as to generate a result set of manageable size, perhaps 1,000 to 2,000 documents.

While a document containing a term of each of the concepts only once is included in a Boolean result set and is possibly only marginally relevant, a document containing frequent appearances of terms of multiple concepts, but completely missing one concept as expressed in the query, is strictly excluded. Additionally, issues such as typographic or indexing errors, or use of abbreviations or unanticipated synonyms are more likely to be influential in citation databases than in full-text collections. Ranking solves these problems and allows users to choose consciously how much effort they are willing to invest into the search.

Although extended Boolean retrieval has been shown to improve retrieval results compared to strict Boolean evaluation, the complexity in specifying a Boolean query does not pay off if similar retrieval results can be achieved

with ranking and simple keyword queries. However, previous work in the medical domain has shown that ranked retrieval using the information need descriptions at hand did not lead to higher performance [Karimi et al., 2009]. One possible explanation is that the complex information needs in this domain cannot be expressed as bag-of-words queries amenable to currently used ranking functions.

2.2 Related Work

For years, the use of Boolean queries has been deeply embedded in process guidelines for biomedical systematic review search [Higgins and Green, 2008]. As a consequence, there is much literature investigating issues around them. Dickersin et al. [1994] found that, although the sought documents are present in MEDLINE, the sensitivity of queries is unsatisfactory even if only metadata is searched. Beahler et al. [2000] conclude that Boolean search is not enough because binary matching is insufficient due to indexing errors. Also, search in multiple databases has been suggested to alleviate this problem [Avenell et al., 2001].

Few papers suggest ranking as a solution to the low answer relevance density of Boolean queries. Martinez et al. [2008] use different textual information from the systematic review as a query to get an initial ranking which is then refined using a pseudo-relevance feedback technique. Karimi et al. [2009] found that loosening the strictness of the queries combined with ranking of the result set is able to achieve higher relevance fractions and outperform each individual method. Still, the low relevance densities attained suggest that for medical abstract searching the highest realistic aim can only be to find as much relevant literature as possible for a given effort, rather than finding every relevant document. However, using textual descriptions as keyword queries with typical ranking functions has recently been shown to perform poorly [Bendersky and Croft, 2008]. It is thus surprising that descriptive ranked queries perform as well as Boolean queries, although key concepts are formally present in the Boolean query specifications. Although the previous work considered ranking, none of the studies used the Boolean queries themselves for this purpose.

Cohen et al. [2006] estimate the usefulness of an approach based on classification of the citations in the Boolean result set to screen out documents that are likely to be irrelevant. To train a classifier, they assumed half of the documents to be judged. Naturally, this limits the possible improvement. While this approach might be able to reduce costs associated with judging documents, it cannot find additional relevant document not in the Boolean result set. However, if applied to filtering newly published documents for their relevance to systematic reviews, more relevant documents could be found than with Boolean filters. Shojania et al. [2007] give evidence that a large fraction of systematic reviews need to be regularly updated.

Extended Boolean models generate ranked output from Boolean query specifications. In the past, many extended Boolean models have been suggested that vary in the ranking function used, the arity of the operators, and in support for query weights. Simple fuzzy-set models [Radecki, 1979] seek to extend the pure Boolean model to support non-binary term weights, but effectively use the same score function as the pure Boolean model. More sophisticated functions are used in the Waller-Kraft [Waller and Kraft, 1979], Paice [Paice, 1984], p-norm [Salton et al., 1983] and Infinite-One models [Smith, 1990]. Lee [1995] gives a good overview of these models.

The sparseness and age of the literature suggests that extended Boolean models have been not as successful as ranking functions based on keyword queries. This seems plausible if similar or better results can be achieved with simple keyword queries which are easier to create. However, it is not clear that this holds in the biomedical domain where complex information needs are to be satisfied that partly include strict inclusion criteria based on meta-data.

2.3 Extended Boolean Retrieval Models

Retrieval models can be characterized based on the assumed query and document representations, the retrieval function, and the form of output. While Boolean models are set-oriented and retrieve only documents that satisfy a Boolean constraint, vector-space models consider documents and queries as vectors. In the latter, both documents and queries are represented as *bags-of-words*, and real-valued similarities are calculated between them and used to rank the documents, a useful presentational device. The former allows representation of more complex information needs in the query. We choose to elaborate on the extended Boolean retrieval model of Salton et al. [1983], also known as p-norm model, selected as a basis for exploration because the ranking formula of this model has been shown to have desirable properties that promote good rankings [Lee, 1994].

To deal with the nested structure of Boolean queries, ranking functions for Boolean queries are defined for each operator separately, working with a tree structure that reflects the structure of the original Boolean expression. The final similarity score is calculated recursively, working from the leaves back to the root of the query tree, applying the corresponding score aggregation formula at each internal node.

The basic operators in each Boolean query are conjunctions (AND) and disjunctions (OR). Salton et al. [1983] define the ranking score s for n-ary disjunctions as

$$s_{\texttt{OR}}(w_1, \ldots, w_n) = \left(\frac{1}{n} \sum_{i=1}^{n} w_i^p \right)^{1/p} ,$$

and for n-ary conjunctions as

$$s_{\texttt{AND}}(w_1, \ldots, w_n) = 1 - \left[\frac{1}{n} \sum_{i=1}^{n} (1 - w_i)^p \right]^{1/p} ,$$

where the w_i values are the weights of terms in the interval $[0, 1]$ when the children are leaves of the tree, or are the scores of the sub-trees also in the range $[0, 1]$, as appropriate to the tree structure below this node; and where p is a parameter in the interval $[1, \infty)$.

The simplest initial choice for document term weights is to assign binary weights at the leaves of the tree, with 0 indicating the absence and 1 the presence of a term. More complex weightings are possible for document as well as the query terms. The choice made for parameter p then determines a particular ranking function within a continuum. In particular, when $p = 1$, a simple inner product document-query similarity function is used, and when $p = \infty$, depending on the choice of term weights, either fuzzy-set or strict Boolean retrieval is performed. If the operand scores are regarded as weights of a vector and defined not to be negative, then the formulas become p-norms of vectors based on the operand scores, normalized to the interval $[0, 1]$. (Hence the name of the technique.)

In pure Boolean systems, it is sufficient to implement conjunctions and disjunctions as binary operators because associativity, op(a,op(b,c)) $\equiv$ op(op(a,b),c), allows n-ary versions of the operator to be handled via any application of multiple binary operators, op(a,b,c) $\equiv$ op(a,op(b,c)). However, in the extended Boolean system associativity does not hold, and terms at higher levels in the operator

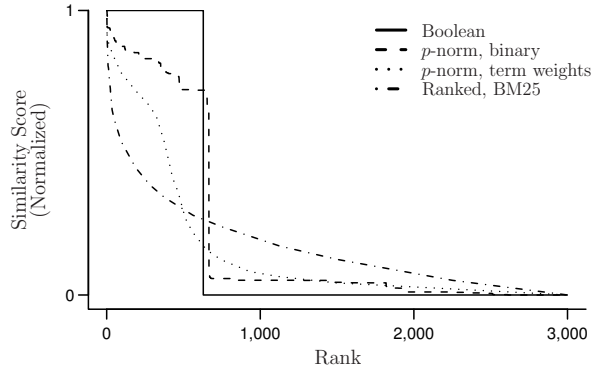

Figure 2: Ranked similarity score distribution for one example query through to rank 3,000, with scores normalized to $[0, 1]$. Note that the domain is actually discrete, and that the score levels at different ranks have been connected for presentational purposes only .

tree have a larger influence on the final query-document similarity score than do lower ones. To circumvent this problem, systems should implement n-ary versions of the operators [Lee, 1994], and queries should be written to make appropriate use of them.

Depending on the number of possible initial term weights, and on the structure of a query, only a limited number of output levels are possible. That is, the degree in which ranking is performed is restricted, and an ordered list of unordered sets is returned, rather than a continuous ranking. This becomes apparent looking at similarity scores of queries for different models, as illustrated in Figure 2. To break similarity score ties a second order criterion can be used, typically the same criterion used for the Boolean result set, for example, reverse chronological. In this sense, continuous term weights might be preferable, presuming some principled method for assigning them can be determined. On the other hand, a study of the score distribution in conjunction with the query structure can be valuable in determining document quantities to which specific operands, or a particular number of conjuncts and disjuncts on the same query tree level match. Recall that the structure of the queries at hand and the p-values used influence the final similarity score, with operators on higher levels in the query tree dominating the score computation.

An interesting observation follows from the example depicted in Figure 2. Based on query-document similarity scores, documents might be clustered into two groups, those with high and low similarity scores. Overall, extended Boolean retrieval using binary term weights tends to follow loosely the shape of the Boolean retrieval function. Thus, the large step in the consecutive similarity scores derives from documents that either satisfy all or only some conjuncts in the conjunction at the top of the query tree. In these two groups, the number of matching disjunctions then determines variation in the scores. However, note that the score gap does not happen at exactly the same rank as for Boolean evaluation, meaning that some documents indeed are able to compensate for less influential, unsatisfied conjuncts. It would be wasteful to exclude these few documents from consideration, because they can be very informative in order to refine the query based on relevancy of these documents.

3 Experiments

We chose to use `Lucene`[4] as indexing engine due to the available `MEDLINE` parser as part of the `LingPipe` project[5]. We then implemented our own query parser, query operators and query evaluators that directly access the inverted lists, assign scores to documents, and return either a set or a ranking of the highest scoring documents. For expansion we used a recent version of `MeSH` as of 2008. Boolean result sets are returned in reverse chronological order because this is the default ordering criterion imposed by most Boolean search systems. That is, within a set of equal-score items, the recent ones are returned preferentially over older ones, on the assumption that age degrades usefulness.

We compare our results against two Boolean baselines, our own implementation, and `Ovid`, the interface to which these medical queries are usually submitted. We entered the queries into this online system, and retrieved the `PubMed` identifiers of each returned document. Furthermore, analogous to the work presented by Martinez et al. [2008], we used `zettair` (version 0.9.3) with the Okapi ranking function [Robertson et al., 1995] and default settings, to generate keyword query baselines based on review title, research questions, inclusion criteria and the plain terms used in Boolean queries. These queries are denoted as the TRC formulation in our experiments.

3.1 Data

Relevance judgments and queries for 15 systematic reviews have been made available by Cohen et al. [2006]. Documents are considered to be relevant if they were included in the final review, without regard to how they were found (that is, irrespective of whether they were found by querying the collection, by following citations, or by hands-on involvement via conference proceedings or personal knowledge). This reduces bias in favor of the Boolean queries used. The relevance judgments in this dataset have been restricted to `MEDLINE` documents published between 1993 and 2003 (inclusive) because they were the ones originally made available via the `TREC` Genomics Track [Hersh et al., 2004].

As a collection, we took all active (not withdrawn) abstracts of `MEDLINE` as of late 2008, and a subset consisting of 4,515,866 abstracts published in the year range corresponding to the relevance judgments. We then removed from the relevance judgments any references to documents not in the collection that we had formed. This reduced the number of relevant documents by around 10%, on average; a small penalty in order to create a test environment in which (if it could but exist) a "perfect" query was one which located all of the known relevant documents. Additionally, we used the queries and relevance judgments for 13 additional systematic reviews from a testset made available by Karimi et al. [2009] as a hold-out set, to validate our findings.

3.2 Reproducing results

A key requirement in this domain is reproducibility, which is why the Boolean queries are published together with the reviews that they contribute to. However, it has been noted that the published queries contain typographic errors [Sampson and McGowan, 2006]. Additionally, metadata such as `MeSH` headings are used which are under continuous refinement. Fortunately, many authors also document the result set sizes achieved for each component line in their query script, which could be used as a guide during query debugging. Even so, differences in document parsing, in the actual document sets used, in query parsing and transformation (for instance term mapping or error correction), means that there can be variation between systems executing the same Boolean query. This might be one of the reasons why it is recommended that multiple databases are to be searched, even though there is considerable overlap between databases.

[4] `http://lucene.apache.org/java/`
[5] `http://alias-i.com/lingpipe/`

Query	# Nodes	Result set size		
		Ovid	Reprod.	Overlap
1	66	1,597	1,457	1,452
2	35	979	884	884
3	120	353	342	342
4	71	735	629	629
5	47	3,856	3,805	3,727
6	37	1,377	1,340	1,337
7	77	306	248	248
8	38	286	271	271
9	80	742	823	629
10	26	282	258	258
11	57	1,108	1,106	1,106
12	72	718	682	682
13	44	2,460	2,331	2,104
14	51	413	372	372
15	409	823	801	801
Mean	112	1,069	1,023	989

Table 2: Number of query tree nodes, and Boolean result set sizes using Ovid and our own system (column "Reprod.") for each of fifteen test queries. The final column shows the cardinality of the intersection between the two results sets.

We fixed all obvious errors in the queries (such as unmatched terms), and changed MeSH headings that had been subsequently refined so that they referred to the appropriate subheadings, and reached the point where we reasonably reproduce results that we got by using Ovid directly with our own system (Table 2). Note that the differing result set sizes are typical of the issues with Boolean queries. The searches using Ovid were performed in mid 2009 and any documents not in our collection have been removed from those result sets.

3.3 Evaluation measure

The typical evaluation measures used in IR are precision and recall, measured either across the whole of a Boolean result set, or at some particular depth in the ranking, if the system assigns scores to documents. In order to incorporate more information about the distribution of relevant documents over ranks, more complex evaluation measures can also be used for rankings, for example, MAP, NDCG or RBP (see Moffat and Zobel [2008] for a summary of these). However, all of these measures place considerable emphasis on the top of the ranking; and in the medical domain all documents in the final result set are likely to be inspected for relevancy. Hence, a set-based measure is to be preferred.

In the particular application under consideration, very patient searches are undertaken, and costly evaluation processes are tolerated in the interests of research quality. It is thus not unusual for thousands of documents to be inspected once a query has been finalized, and whereas typical web-search quality can be regarded as being measured by (say) precision over the first five documents in the ranking, here we wish to evaluate a system according to whether, over a substantial answer set, a comprehensive set of answers has been located. In addition, that evaluation measure should not pay terribly much regard to exactly where in the ranking (or answer set) the relevant documents occur.

Precision is easy to measure, since it is based on the documents encountered when traversing through an answer set or ranked list. On the other hand, recall (at least, in its formal definition, see Zobel et al. [2009] for discussion of recall-like metrics) is retrospective, and can only be computed after all of the relevant documents in a collection have been identified. If exhaustive inspection of documents is impossible (and in all realistic scenarios that is the situation), any estimate as to the number of relevant documents has some element of uncertainty associated with it. It is certainly true that if many different techniques are applied independently and the results pooled, then a greater number of relevant documents are likely to be found, and hence the number of relevant documents not found must have decreased. But even so, the best that can be said is that the total number of relevant documents that have been identified provides both a lower bound on the actual number of relevant documents, and an upper bound on the number of relevant documents that any particular system can, in an experiment, be expected to identify.

In the experiments reported below, we measure the number of relevant documents found at different ranks based on the size of the Boolean result set and for different absolute ranks, respectively; and then standardize them across queries by dividing by the number of known relevant documents for each query before computing an overall average. However, we refrain from denoting this quantity as being "recall", since the entire document collection has not been inspected, and actual recall scores will thus likely be lower than the values reported. Note also that most valuable for systematic review search is to find more relevant documents in the (say) top 500 to $2{,}000$ ranks. The Boolean queries are likely to have been targeted to return up to that many documents, because at these depths in the ranking the number of documents to inspect is of manageable size.

3.4 Implementation

We replaced higher order operators, such as explosions of MeSH headings by disjunctions of all headings below that entry in the MeSH hierarchy; terms containing wildcards by disjunctions of all terms in the dictionary that match the given pattern; and MeSH entries themselves by term lookups in a particular field containing the MeSH headings of the documents.

All complex operators were mapped to the three basic Boolean operators in order to execute them in an extended Boolean fashion. Because we wanted to profit from the extended Boolean interpretation of all restrictive operators, we chose to do this also with phrases and proximity operators. They became simple conjunctions and thus allowed documents to still be ranked highly if one term of a phrase did not occur in the document. For efficiency reasons, MeSH explosions and wildcard term expansions have always been implemented as Boolean disjunctions. This did not affect retrieval effectiveness because often the expansions are spurious terms due to spelling or parsing errors that occur only in a few documents but increase query size excessively. Otherwise, they are flection variations, synonymous terms or hyponyms and should be treated equally if the hypernym is queried.

The original queries were not intended to be executed in an extended Boolean sense and hence, no care has been taken to specify Boolean connectives in their binary or n-ary form – the associativity property of Boolean logic means that any equivalent specification leads to the same result. But this does not hold for the ranking functions used in extended Boolean models. Here, the structure of the query tree determines the influence of operands and thus the final similarity score. To normalize the queries making them more amenable to extended Boolean evaluation, we logically flattened cascades of (often binary) conjunctions or disjunctions, respectively, into their n-ary equivalents. For instance, a query

```
(a or (b and c)) and (d and (e or (f or g)))
```

would be transformed to

```
and(or(a, and(b, c)), d, or(e, f, g))
```

Id	System	Number of relevant documents (normalized)								
		at absolute ranks					at ranks relative to B_q			
		100	300	1,000	3,000	10,000	$0.25B_q$	$0.5B_q$	B_q	$2B_q$
a:	p-norm$_{\text{BIN}}$, $p=9$	0.18^{cd}	$0.40^{c'd'e}$	$0.62^{c'd'e}$	$0.72^{c'd'}$	$0.79^{c'd'}$	$0.28^{cd'}$	0.44^{de}	$0.61^{be'}$	$0.69^{b'c'd'e}$
b:	p-norm$_{\text{TF·IDF}}$, $p=9$	0.16^{c}	0.34	0.57	0.70^{c}	$0.77^{c'd'}$	0.22	0.43	0.59^{e}	0.63
c:	Boolean Reprod.	0.09	0.29	0.48	0.58	0.59	0.16	0.34	0.59^{e}	
d:	Boolean Ovid	0.10	0.27	0.49	0.60	0.61	0.14	0.31	0.58^{e}	0.61
e:	TRC queries	0.15	0.27	0.48	0.66	$0.81^{c'd'}$	0.20	0.29	0.41	0.58

Table 3: Number of relevant documents found for different systems, averaged over 15 AHRQ queries and normalized by the number of known relevant documents. We measured at absolute ranks and ranks based on multiples of B_q, the size of the reproduced Boolean result set for query q. We also tested for statistical significant differences between all systems using a one-sided Wilcoxon signed-rank test and report improvements at the 0.05 (Id) and 0.01 (Id′) levels, respectively.

Id	System	Number of relevant documents (normalized)								
		at absolute ranks					at ranks relative to B_q			
		100	300	1,000	3,000	10,000	$0.25B_q$	$0.5B_q$	B_q	$2B_q$
a:	p-norm$_{\text{BIN}}$, $p=9$	0.10	0.22	$0.36^{c'}$	$0.45^{c'd}$	$0.54^{b'c'd'}$	0.19	0.26	0.34	$0.40^{c'}$
b:	p-norm$_{\text{TF·IDF}}$, $p=9$	0.08	0.17	0.31	0.41	$0.47^{c'd}$	0.12	0.24	0.34	0.37
c:	Boolean Reprod.	0.06	0.15	0.27	0.36	0.36	0.13	0.22	0.36^{de}	
d:	Boolean Ovid	0.05	0.16	0.29^{ac}	0.37	0.37	0.13	0.21	0.30	0.37
e:	TRC queries	0.11	0.19	0.30	0.40	0.48^{c}	0.16	0.21	0.27	0.33

Table 4: Performance on a hold-out set of 13 AHRQ queries.

3.5 Experimental setup

As a ranked retrieval baseline, we reproduced the results reported by Martinez et al. [2008], and confirmed their findings. More evidence as part of the query lead consistently to better results. Thus, we only report the results for TRC queries that are based on the concatenation of title, research question and inclusion criteria. These performed significantly better than queries using only part of the information, or the terms in the Boolean queries.

For the p-norm model, we tried a range of p-values suggested in the literature. The best performance was recorded with $p=9$, but retrieval effectiveness was not very sensitive to this parameter. At the extremes, use of $p=\infty$ returned the same results (using binary term weights) as obtained with a strict Boolean implementation, and $p=1$ performed worse than using a modern keyword ranking function naively on the terms in the Boolean query. We consider binary term/leaf weights as input to the p-norm computation; and also term/leaf weights based on the TF·IDF formulation proposed by Salton et al. [1983].

3.6 Results

Table 3 summarizes the results, where each reported value is the fraction of the known relevant documents determined by the given ranking depth, averaged over the query set. The superscript letters give the result of significance tests against other methods listed in the table. For example, the second entry in the table's first row, $0.40^{c'd'e}$, indicates that on average 40% of the known relevant documents are within the top 300 answers for the p-norm method, and that this is significantly better at the 0.01 level than the averages in the same column in rows c and d of the table, and significantly better at the 0.05 level than the average value listed in row e of the table.

First, note that our Boolean baseline system (denoted "Reprod.") gives performance similar to the `Ovid` results, further confirming the accuracy of our baseline.

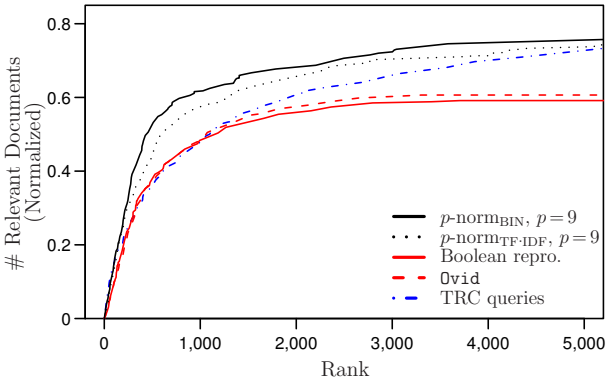

Figure 3: Average number of relevant documents (normalized by number of known relevant documents) for 15 AHRQ queries as a function of ranking depth.

Second, if absolute ranks are considered (the left-hand half of the table), the extended Boolean system with binary term/leaf weights always finds significantly more relevant documents than the other methods. This is due to averaging over multiple Boolean queries that have varying output sizes. While queries with large Boolean result sets have an increased density of relevant document at the beginning of their ranking, queries with small result sets might not be able to produce that many documents. This exemplifies the advantage of ranking, but does not compare the performance of the system for each individual query.

The third observation is drawn from the right-hand half of Table 3, which reports results at answer set sizes calculated as multiples of the size of the result sets for the original Boolean queries, rather than at fixed depths. When a quarter of the Boolean set size is inspected, on average double that many relevant documents can be found using the p-norm model with binary term weights.

This suggests the benefit of ordering Boolean result sets by relevance rather than recency. Although we could not show any improvement when the same number B_q of documents for each query are inspected, we find significantly more documents when double that many documents are viewed. Because result set sizes have not consciously been chosen and might thus be smaller than the capacities of the review team allow for, it seems to be a valuable investment to be able to look behind the cut-off that pure Boolean execution enforces.

Fourth, while the broader, ranked TRC queries are able to catch up at retrieval sets of size $5{,}000$, inspecting that many documents is likely to be infeasible for small review teams, and other retrieval techniques would be preferred (compare Figure 3). Weighting terms individually in the extended Boolean system could not compete with binary weights though. This may be a result of inappropriate weight assignments, and is a direction we plan to explore further – it seems counter-intuitive that starting with binary term/leaf values can be superior to starting from real-valued ones, provided those values are well chosen.

Overall, we conclude that the extended Boolean retrieval results are significantly better than the `Ovid` ones in almost all situations.

Because we developed our system with the test queries and tuned the p-value to that set of queries, we also generated a new testset containing 13 further queries, and applied the same experiment. Table 4 gives the outcomes. The queries in this dataset identify smaller fractions of the known relevant documents, and there are fewer significant differences. But the same overall trends are apparent, confirming our findings.

4 Conclusions

In the biomedical domain with its complex information needs, it turns out that Boolean querying is on a par with ranked approaches using TRC queries built up from review title, research question, and inclusion criteria. Extended Boolean retrieval models are able to increase the fraction of relevant documents found after inspecting the usual 500 to $2{,}000$ documents, by loosening the strictness of conjunctive operators and introducing some elements of ranking. This flexibility allows users to consciously choose the investment they are willing to make in inspecting answers. In our experiments, a simple extended Boolean retrieval model with binary term weights outperformed pure Boolean and ranked retrieval. Finding more documents early in the search process reduces the risk of biasing the outcome of the later employed discovery techniques, such as following citation links of already found publications or asking their authors.

If Boolean result sets are required, the proposed approach can be combined with strict Boolean querying at least in the following two approaches. First, ordering Boolean result sets by extended Boolean scores allows assessment of the quality of the query in support of iterative query refinement without sampling or inspecting the whole set. Second, after the Boolean set is reported, it can be extended by any number of documents. Either as-yet reported documents are returned by descending similarity score, or successively additional sets can be returned that do not match on (say) one or two conjuncts, or conjuncts that have the least impact on overall retrieval score. As is also the case with Boolean retrieval, binary term weights have the advantage that only the document itself determines its similarity score, not being dependent on properties of all documents in the collection. This is helpful to reproduce results and independent determination why a particular document has not been found with the published search strategy.

5 Future Work

Specific query parts should, by definition, be executed in a strict Boolean sense because they reflect inclusion criteria and are often based on (presumably) more reliable metadata rather than free-text. More generally, this reduces to using different p-norms for different query operator types as well as individual query operators. Further improvement might also be possible if adapted ranking functions could be found for each of the used operators. For example, we are currently ignoring information by treating phrases as conjunctions.

Continuous or diversified scores allow the estimation of a cut-off level that could be used to filter newly published documents for their relevancy. Extended Boolean retrieval could be used as an alternative to a strict Boolean filter. However, the weighting scheme that we applied to document terms has been very basic and was demonstrated to be inferior to binary weights. Further improvements are likely if better weights can be assigned to document and query terms, for instance, conditioned on the results of other query parts. Also, our keyword queries are likely to still be suboptimal, but it is not clear how to get to better queries in this context.

Finally, ranking complexity is generally linear in the size of the query and the number of documents matching at least one term. The longer the query the more likely this becomes to be the whole collection. While many optimizations have been applied to ranking of keyword queries, we are unaware of efficient implementations for extended Boolean retrieval models, and plan to explore this issue as the next step in this project.

Acknowledgements National ICT Australia (NICTA) is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

References

Alison Avenell, Helen H. Handoll, and Adrian M. Grant. Lessons for search strategies from a systematic review, in The Cochrane Library, of nutritional supplementation trials in patients after hip fracture. *The American Journal of Clinical Nutrition*, 73(3):505–510, March 2001. PMID: 11237924.

Chris C. Beahler, Jennifer J. Sundheim, and Naomi I. Trapp. Information Retrieval in systematic reviews: Challenges in the public health arena. *American Journal of Preventive Medicine*, 18(4 Suppl):6–10, May 2000. PMID: 10793275.

Michael Bendersky and W. Bruce Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–498, Singapore, July 2008.

Aaron M. Cohen, William R. Hersh, K. Peterson, and Po-Yin Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, 13(2):206–219, March 2006. PMID: 16357352.

Frank Davidoff, Brian Haynes, Dave Sackett, and Richard Smith. Evidence based medicine. *British Medical Journal*, 310(6987):1085–1086, April 1995.

Kay Dickersin, Roberta Scherer, and Carol Lefebvre. Systematic Reviews: Identifying relevant studies for systematic reviews. *British Medical Journal*, 309 (6964):1286–1291, November 1994. PMID: 7718048.

Valery I. Frants, Jacob Shapiro, Vladimir G. Voiskunskii, and Isak Taksa. Boolean search: Current state and perspectives. *Journal of the American Society for Information Science*, 50(1):86–95, January 1999.

William Hersh. *Information Retrieval: A Health and Biomedical Perspective*. Springer, 3rd edition, November 2008.

William R. Hersh, Ravi Teja Bhupatiraju, Laura Ross, Aaron M. Cohen, Dale Kraemer, and Phoebe Johnson. TREC 2004 Genomics Track overview. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*, pages 16–19, Gaithersburg, Maryland, USA, November 2004. NIST. Special Publication 500-261.

J. P. T. Higgins and S. Green, editors. *Cochrane Handbook for Systematic Reviews of Interventions*. Version 5.0.2 [updated September 2009]. The Cochrane Collaboration, 2008. Available from `http://www.cochrane-handbook.org`.

Sarvnaz Karimi, Justin Zobel, Stefan Pohl, and Falk Scholer. The challenge of high recall in biomedical systematic search. In *ACM 3rd International Workshop of Data and Text Mining Methods in Bioinformatics (DTMBIO '09)*, November 2009.

Joon Ho Lee. Properties of extended Boolean models in Information Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 182–190, Dublin, Ireland, 1994. Springer-Verlag New York, Inc.

Joon Ho Lee. Analyzing the effectiveness of extended Boolean models in Information Retrieval. Technical Report TR95-1501, Cornell University, 1995.

David Martinez, Sarvnaz Karimi, Lawrence Cavedon, and Timothy Baldwin. Facilitating biomedical systematic reviews using ranked text retrieval and classification. In *Proceedings of the 13th Australasian Document Computing Symposium (ADCS '08)*, pages 3–10, Hobart, Tasmania, Australia, December 2008.

Jessie McGowan and Margaret Sampson. Systematic reviews need systematic searchers. *Journal of the Medical Library Association*, 93(1):74–80, January 2005. PMID: 15685278.

Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems*, 27(1):1–27, December 2008.

Chris D. Paice. Soft evaluation of Boolean search queries in Information Retrieval systems. *Information Technology: Research and Development*, 3(1):33–41, January 1984.

Tadeusz Radecki. Fuzzy set theoretical approach to document retrieval. *Information Processing & Management*, 15(5):247–259, 1979.

Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Mike Gatford, and Alison Payne. Okapi at TREC-4. In *The 4th Text REtrieval Conference (TREC-4)*, pages 73–96, Gaithersburg, Maryland, USA, November 1995. NIST. Special Publication 500-236.

Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean Information Retrieval. *Communications of the ACM*, 26(11):1022–1036, November 1983.

Margaret Sampson and Jessie McGowan. Errors in search strategies were identified by type and frequency. *Journal of Clinical Epidemiology*, 59(10):1057.e1–1057.e9, October 2006. PMID: 16980145.

Margaret Sampson, Jessie McGowan, Jennifer Tetzlaff, Elise Cogo, and David Moher. No consensus exists on search reporting methods for systematic reviews. *Journal of Clinical Epidemiology*, 61(8):748–754, August 2008. PMID: 18586178.

Kaveh G. Shojania, Margaret Sampson, Mohammed T. Ansari, Jun Ji, Steve Doucette, and David Moher. How quickly do systematic reviews go out of date? A survival analysis. *Annals of Internal Medicine*, 147(4): 224–233, August 2007. PMID: 17638714.

Maria E. Smith. *Aspects of the P-Norm model of Information Retrieval: Syntactic query generation, efficiency, and theoretical properties*. PhD thesis, Cornell University, May 1990.

W. G. Waller and Donald H. Kraft. A mathematical model of a weighted Boolean retrieval system. *Information Processing and Management*, 15(5):235–245, 1979.

Li Zhang, Isola Ajiferuke, and Margaret Sampson. Optimizing search strategies to identify randomized controlled trials in MEDLINE. *BMC Medical Research Methodology*, 6(1):23, May 2006. PMID: 16684359.

Justin Zobel, Alistair Moffat, and Laurence Park. Against recall: Is it persistence, cardinality, density, coverage, or totality? *SIGIR Forum*, 43(1):3–15, June 2009.