ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Information Retrieval by Enhanced Boolean Model

K.Madhuri[1], L.Divya[2]

Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, Telangana, India[1]

Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, Telangana, India[2]

ABSTRACT*:* The interest for information retrieval has existed long before the Internet. Boolean retrieval is the most simple of these retrieval methods and relies on the use of Boolean operators. The terms in a query are linked together with AND, OR and NOT. This method is often used in search engines on the Internet because it is fast and can therefore be used online. This method has also its problems. The user has to have some knowledge to the search topic for the search to be efficient, e.g., a wrong word in a query could rank a relevant document non relevant. The retrieved documents are all equally ranked with respect to relevance and the number of retrieved documents can only be changed by reformulating the query. We consider p-norm approach, Max score and wand exact optimization techniques for ranked keyword retrieval that can be adopted via low cost screening process.

KEYWORDS*:* Information Retrieval, Boolean Model, Ranked Keyword, Vector Space, scoring methods.

I. 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 arewilling 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.

DOI: 10.15680/IJIRSET.2014.0311022

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

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. 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.

II. BACKGROUND

Boolean Retrieval models produce meaningful rankings, their query model allows the representation of complex concepts in an and-or format; and they are scrutable, in that the score assigned to a document depends solely on the content of that document, unaffected by any collection statistics or other external factors. These characteristics make Boolean Retrieval models attractive in domains typified by medical and legal searching, where the emphasis is on iterative development of reproducible complex queries of dozens or even hundreds of terms. However, Boolean Retrieval is much more computationally expensive than the alternatives. We consider the implementation of the p-norm approach to Boolean Retrieval, and demonstrate that ideas used in the max-score and wand exact optimization techniques for ranked keyword retrieval can be adapted to allow selective bypass of documents via a low-cost screening process for this and similar retrieval models. The proposed term independent bounds that are able to further reduce the number of score calculations for short, simple queries under the extended Boolean retrieval model. Together, these methods yield an overall saving from 50 to 80 percent of the evaluation cost on test queries drawn from biomedical search.

Ranking Algorithms: Max-score ranking mechanism, to accelerate keyword query evaluation when sum-score aggregation functions are used and only the top-k documents are required. Using document-at-a-time evaluation, the algorithm commences by fully scoring the first k documents in the OR-set of the query terms. Thereafter, the kth largest document score is tracked, as an entry threshold that candidate documents must exceed before they can enter the (partial) ranking. The max-score algorithm uses the information conveyed by the entry threshold to reduce two cost factors: 1) the number of candidate documents that are scored; and 2) the cost associated with scoring each candidate document. To achieve this, the terms in the ranked query are ordered by decreasing document frequency. Then, for each term ti in the ordering, the highest achievable score is computed for a document containing all of the terms t1 _ _ _ ti. To compute the threshold score for tiþ1, the maximal term-contribution of tiþ1 is determined, and then added to the score of ti. During query processing, the entry threshold is monotonically increasing, and at some point is likely to become sufficiently large that it can be concluded that a document containing only the commonest term t1 (and none of the other terms) cannot make it into the top k. At that moment in time the set of candidate documents to be checked is reduced to the OR-set of t2; . . . ; tn, and processing continues until the threshold associated with t2 is also less than the entry threshold. Independently, these score bounds also allow short-circuiting of the evaluation of each candidate document, so that not all terms are necessarily inspected. Note that the document frequency dictates the order in which terms are evaluated.

Algorithm1: Query tree Scoring
Input: Consider,
T = a set of numbered terminals,
B = a set of numbered internal nodes,
N = a set of tree nodes describing a Boolean expression.
Step1: S = set of query terms.
$S \leftarrow \{ T_i \in T \mid T_i(S) > 0 \}$.
Step2: While $S \neq \{N_1\}$ do

DOI: 10.15680/IJIRSET.2014.0311022

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Step3: Determine largest parent Node index "j".
Step4: Determine active clauses "A" of B_j in S.
Step5: Split A into two sets
A € (S=1) and A € (0<S<1).
Step6: if A € (0<S<1) = 0 then
Step7: Lookup pre-computed score when operands are all binary.
Step8: end.
Step9: Remove the processed nodes from S, and add their parent:
S←S - A + $\{B_j\}$
Step10: end.
Step11: return $N_1(S)$.

An alternative to tracking the score of the kth document is to specify a minimal entry threshold prior to any documents being scored, making the retrieval of documents reproducible, but meaning that the result set for any given query will grow as the collection grows. Or, if term contributions differ between documents, then a higher initial threshold can be attained when top-scoring documents for each term are precomputed and stored as additional lists at indexing time, and then merged for each query before query evaluation starts. Here demonstrate that these methods do indeed reduce the number of documents that have to be scored, and that retrieval times were also improved. Note, however, that to date these methods have primarily been applied to ranking of flat keyword queries, possibly extended with proximity operators and phrases, and that they have not been applied to structured queries because the overall scoring functions do not decompose into a sum over term contributions.

III. OPTIMIZATION TECHNIQUES

Optimization Principles: Inverted lists are generally accepted to be the most appropriate data structure for information retrieval systems, with the two main query evaluation strategies being term-at-a-time and document-at-a-time. In the former, the inverted list of each query term is fully processed before the next is opened, and intermediate scores for all candidate answers are stored in a set of accumulators. This leverages fast sequential disk access, and when only one value has to be stored per candidate document as is the case with ranked queries is an attractive approach. In addition, the number of accumulators can be restricted without any great loss in effectiveness. However, when complex Boolean queries are being processed, each accumulator is a complex structure corresponding to the complete state of a partially processed query. Nor can pruning be used to reduce the cost, because the set of answers to be produced is deterministic, rather than heuristic.Document-at-a-time processing accesses all of the inverted lists at the same time, stepping through them concurrently and fully considering any document that appears in any of the lists before moving on to the next. Note that, because of compression considerations, inverted lists are typically ordered by document number, so document-at-a-time systems operate as a type of multiway merge, and do not need to backtrack through the lists. This simplifies the implementation of nested and complex operators, and there is no storage of intermediate results for any documents except the current one. The drawback is that the resultant data access pattern is multi location sequential rather than single-location sequential and explicit or implicit (by allowing the operating system to prefetch blocks of data) buffering must be used, so that the great majority of "get next document pointer" operations are still performed out of memory.

The vector space model: The vector space model procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The last stage ranks the document with respect to the query according to a similarity measure.The vector space model has been criticized for being ad hoc.

Document Indexing: It is obvious that many of the words in a document do not describe the content, words like *the, is*. By using automatic document indexing those non significant words (function words) are removed from the document vector, so the document will only be represented by content bearing words. This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words. In practice, term frequency has been difficult to implement in automatic indexing. Instead the use of a stop list which holds common words to remove high frequency words (stop words), which makes the indexing method language

DOI: 10.15680/IJIRSET.2014.0311022

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

dependent. In general, 40-50% of the total numbers of words in a document are removed with the help of a stop list.Non linguistic methods for indexing have also been implemented. Probabilistic indexing is based on the assumption that there is some statistical difference in the distribution of content bearing words, and function words .Probabilistic indexing ranks the terms in the collection with respect to. the term frequency in the whole collection. The function words are modelled by a Poisson distribution over all documents, as content bearing terms cannot be modelled. The use of Poisson model has been expand to Bernoulli model . Recently, an automatic indexing method which uses serial clustering of words in text has been introduced .

Term Weighting: Term weighting has been explained by controlling the exhaustively and specificity of the search, where the exhaustively is related to recall and specificity to precision. The term weighting for the vector space model has entirely been based on single term statistics. There are three main factors term weighting: term frequency factor, collection frequency factor and length normalization factor. These three factor are multiplied together to make the resulting term weight. A common weighting scheme for terms within a document is to use the frequency of occurrence as stated by, mentioned in the previous section. The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a weighted document vector. It is also possible to use binary document vector, but the results have not been as good compared to term frequency when using the vector space model .There are used various weighting schemes to discriminate one document from the other.In general this factor is called collection frequency document.

IV. EXPERIMENTAL RESULTS

Generally speaking, any form of ranked query evaluation, including both conventional keyword-based ranking an EBR, takes time that is linear in the product of the number of documents that match at least one term (the OR-set size) and of the query complexity. If a query contains very common terms, the first of these two factors can result in every document in the collection needing to be scored. This is common with complex Boolean queries. Worse, complex queries sometimes contain hundreds of terms, meaning that the second factor can also be high. We investigate ways to reduce both of these costs.

Scoring Method: Rather than being a simple sum, the overall scoring function in the p-norm model is a nested application, determined by the query tree. Hence, it is not possible to aggregate the score for a document starting with any arbitrary term, including the one with the lowest document frequency. The recursive nature of EBR queries makes it necessary to calculate the scores on lower levels in the query tree first. One obvious possibility would be to try and add processing logic to each query node as it acts on its clauses. But optimizations such as max-score could only be employed at the query root node, as a threshold is only available for the overall query score. Instead, we follow a holistic approach and prefer to be able to calculate the document score given a set of query terms S _ T present in a document, no matter where they appear in the query tree. Our approach, described in Algorithm 1, assumes that each query tree node Ni, for i ¼ 1; . . . ; n, is assigned a smaller index identifier than any of its children, so that Ni:P < i, where Ni:P is the index of the parent node of Ni, and where each node N is either a term drawn from T, or a Boolean operator drawn from B. It necessarily follows that N1 denotes the query's root node.

There are at least two possible applications for term independent score bounds. First, they can be used instead of the adaptation of max-score. While max-score imposes an order on the terms in which they are excluded from the OR-set, term-independent bounds are able to dynamically exclude a number of arbitrary terms. When the current entry threshold exceeds all query terms can be (partially) sorted by their next candidate document identifier, and then the first r terms advanced (by a skipping process). Only then is it necessary for a document to be fully scored using CalcScore(), since to enter the answer set a document must (currently) contain more than r of the query terms. This approach is similar to the processing performed in the wand algorithm [19], but we do not calculate score bounds dynamically due to the demanding score calculations.

Second, TIB can be combined with the adaptation of max-score. Even after the max-score approach has indicated that a document needs to be scored, the TIB filter might successfully eliminate that document before all inverted lists are consulted, based purely on how many of the query terms it could contain. For example, by inspection of inverted lists in order of document frequency, it might be known that a candidate document only contains one term

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

plus possibly one other of the query terms that are yet to be consulted. If the current entry threshold score translates into a minimum of three query terms, this document cannot make it into the answer set and can thus be discarded. This saves us from scoring the document, and also increases the skip size on common terms, such as Humans.

V. CONCLUSION

Having noted that ranked keyword querying is not applicable in complex legal and medical domains because of their need for structured queries including negation, and for repeatable and securable outputs, we have presented novel techniques for efficient query evaluation of the p-norm (and similar) extended Boolean retrieval model, and applied them to document-at-a-time evaluation. We showed that optimization techniques developed for ranked keyword retrieval can be modified for EBR, and that they lead to considerable speedups. Further, we proposed term-independent bounds as a means to further short-circuit score calculations, and demonstrated that they provide added benefit when complex scoring functions are used.

Finally, there might be other ways to handle negations worthy of consideration. We also plan to evaluate the same implementation approaches in the context of the inference network and wand evaluation models. For example, it may be that for the data we are working with relatively simple choices of term weights in particular, strictly document-based ones that retain the scrutability property that is so important can also offer good retrieval effectiveness in these important medical and legal applications.

VI. ACKNOWLEDGEMENT

The Successful Completion of our task would be incomplete without expression of simple gratitude to the people who encouraged our work. The words are not enough to express the sense of gratitude towards everyone who directly or indirectly helped in this task. We are thankful to this Organization VASAVI COLLEGE OF ENGINEERING, which provided good facilities to accomplish my work and would like to sincerely thank to our HOD IT, Dr. N. Vasantha madam and faculty members for giving great support, valuable suggestions and guidance in every aspect of my work.

REFERENCES

[1] G. Salton, E.A. Fox, and H. Wu, "Extended Boolean Information Retrieval," *Comm. ACM*, vol. 26, no. 11, pp. 1022-1036, Nov. 1983.
[2] F. McLellan, "1966 and All that—When Is a Literature Search Done?," *The Lancet*, vol. 358, no. 9282, p. 646, Aug. 2001.
[3] S. Karimi, J. Zobel, S. Pohl, and F. Scholer, "The Challenge of High Recall in Biomedical Systematic Search," *Proc. Third Int'l Workshop Data and Text Mining in Bioinformatics*, pp. 89-92, Nov. 2009.
[4] J.H. Lee, "Analyzing the Effectiveness of Extended Boolean Models in Information Retrieval," Technical Report TR95-1501, Cornell Univ., 1995.
[5] S. Pohl, J. Zobel, and A. Moffat, "Extended Boolean Retrieval for Systematic Biomedical Reviews," *Proc. 33rd Australasian Computer Science Conf. (ACSC '10)*, vol. 102, Jan. 2010.
[6] D. Metzler and W.B. Croft, "Combining the Language Model and Inference Network Approaches to Retrieval,"*Information Processing and Management*, vol. 40, no. 5, pp. 735-750, 2004.
[7] T. Radecki, "Fuzzy Set Theoretical Approach to Document Retrieval," *Information Processing and Management*, vol. 15, no. 5, pp. 247-259, 1979.
[8] W.G. Waller and D.H. Kraft, "A Mathematical Model of a Weighted Boolean Retrieval System," *Information Processing and Management*, vol. 15, no. 5, pp. 235-245, 1979.
C.D. Paice, "Soft Evaluation of Boolean Search Queries in Information Retrieval Systems," *Information Technology Research Development Applications*, vol. 3, no. 1, pp. 33-41, Jan. 1984.