Minimizing Posting List Traversal*

Edward Hills
University of Otago
Dunedin, New Zealand
ehills@cs.otago.ac.nz

ABSTRACT

Information Retrieval is primarily concerned with searching through a document collection given a query, and returning a set of documents which could be relevant to the users request.

This task usually requires searching through the entire document collection to find which documents may be relevant. In large document collections this can be time consuming and costly. This paper looks at ways to minimize the number of documents that are examined but still being effective in returning the documents most likely to be relevant.

1. INTRODUCTION

Searching through document collections which are large can mean when it comes to query time that all documents that are indexed by one or more of the search terms will be examined. When you have a large document collection such as the TREC terabyte collection, searching every document that is indexed by even a relatively rare term can mean searching through thousands or even millions of documents.

By reducing the amount of documents searched by a particular query you do run the risk of limiting the amount of documents returned which are relevant to the user. Keeping the accuracy of your list returned is a top priority when searching through a minimal amount of documents.

There are a number of techniques in which the document list traversal can be minimized while still maintaining high accuracy. This paper looks at three techniques:

1. Index ordering by query-independent measures, Ferguson and Smeaton
2. The Nearest Neighbour Problem In Information Retrieval, van Rijsbergenand Smeaton
3. Filtered Document Retrieval with Frequency-Sorted Indexes, Persin, Zobel and Sacks-Davis

*A summary of previous papers and a suggestion of new ideas

I will examine document minimization further and discuss the main research questions and contributions of each paper in 2., talk about how these techniques are linked in 3., then propose two new ideas in 4. and discuss my final findings in the Conclusion.

2. DOCUMENT MINIMIZATION

By minimizing the document collection that must be evaluated we can avoid unnecessary calculation which slows down performance, lengthens postings list and increases the inverted index file size.

This area of research has been looked into since the early '80s and a multitude of techniques have arisen ranging from Query caching (Lempel and Moran (2003)) and two-tier architectures (Fagni, Perego, Silvestri, and Orlando (2006)) to nearest neighbour searching (Smeaton & van Rijsbergen (1981)) which we talk about lately.

These different techniques are not entirely mutually exclusive as some may think but in some cases can be built upon layer by layer such as the *quit* and *continue* strategies (Moffat & Zobel, 1996) which are used in a variety of different techniques.

With todays web based document collection growing larger and larger the need to weed out documents which are spam or of poor quality to the user needs to be addressed. By removing documents which are of poor quality not only increases the performance of the search engine (as there are less documents to search through given a certain query) but also increases the likelihood that only relevant documents are returned to the user. Removing web documents from a document collection is discussed in the next section.

2.1 Research Questions

2.1.1 Index ordering by query-independent measures

This paper from Ferguson et al. lists a range of heuristics for determining what html documents from the GOV2 test collection from TREC are spam or of poor quality. A common formula for finding good quality documents is PageRank which gives a high priority to documents it thinks is

best. PageRank is used as a query-independent document filter but this paper hopes to do better and find better quality documents. Ferguson et al. come up with some heuristics which include:

- Linkage analysis
- Access counts
- Information-to-noise ratio
- Document cohesiveness
- Document structure and layout
- Click-through data
- Term-specific sorting
- Document length

See [1] for detail about these heuristics.

2.1.2 Paper 2
Briefly talk about the research questions for this paper

2.1.3 Paper 3
Briefly talk about the research questions for this paper

2.2 Main Contributions
Briefly touch on the main contributions for all papers

2.2.1 Paper 1
Describe the main contributions for this paper.

2.2.2 Paper 2
Describe the main contributions for this paper.

2.2.3 Paper 3
Describe the main contributions for this paper.

2.3 Relationship
Talk about how all three papers are related to each other

3. FUTURE WORK
I came up with two ideas in this field which are roughly to do with this... blah blah blah

3.1 Question 1
Well the first question i wanted answered is, can we do this... blah blah blah. the best way to do this would be combine this this and that and then it may be possible to imporve performance or not blah blah blah

3.2 Question 2
Well the first question i wanted answered is, can we do this... blah blah blah. the best way to do this would be combine this this and that and then it may be possible to imporve performance or not blah blah blah

4. CONCLUSIONS
You can see that this paper has described a multitude of different ways in which we can minimize the cost of transfering or merging the queries in a distributed IR environment. by taking points from paper 1 and paper 2 they can be combined and blah blah blah. Paper 3 talks aboue the architecture and we can see that this is important because of blah.

I came up with my own 2 points and found that blah blah blah

5. REFERENCES
5.1 References
Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.