Maximal Termsets as a Query Structuring Mechanism [*] [†]

Bruno Pôssas [1,2]
bavep@dcc.ufmg.br

Nivio Ziviani [1]
nivio@dcc.ufmg.br

Berthier Ribeiro-Neto [1,2]
berthier@dcc.ufmg.br

Wagner Meira Jr. [1]
meira@dcc.ufmg.br

ABSTRACT

Search engines process queries conjunctively to restrict the size of the answer set. Further, it is not rare to observe a mismatch between the vocabulary used in the text of Web pages and the terms used to compose the Web queries. The combination of these two features might lead to irrelevant query results, particularly in the case of more specific queries composed of three or more terms. To deal with this problem we propose a new technique for automatically structuring Web queries as a set of smaller subqueries. To select representative subqueries we use information on their distributions in the document collection. This can be adequately modeled using the concept of maximal termsets derived from the formalism of association rules theory. Experimentation shows that our technique leads to improved results. For the TREC-8 test collection, for instance, our technique led to gains in average precision of roughly 28% with regard to a BM25 ranking formula.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Retrieval models*; H.3.3 [**Information Systems**]: Information Search and Retrieval—*Query formulation*; H.3.4 [**Information Systems**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; H.2.8 [**Information Systems**]: Database Applications—*Data mining*

General Terms

Theory, Algorithms, Experimentation

Keywords

Information retrieval models, association rule mining, weighting index term co-occurrences, automatic query structuring

[*]Full version available as a Technical Report (TR012/2005), Dept. of Computer Science, Federal Univ. of Minas Gerais, Brazil, http://www.dcc.ufmg.br/∼nivio/papers/tr012-2005.pdf.

[†]This work was supported in part by CNPq scholarship 141.269/02-2 (Bruno Pôssas), by CNPq grant 520.916/94-8 (Nivio Ziviani), by CNPq grant 30.0188/95-1 (Berthier Ribeiro-Neto), by CNPq grant 30.9379/03-2 (Wagner Meira Jr.), and by the GERINDO project grant MCT/CNPq/CT-INFO 552.087/02-5.

[1] Federal University of Minas Gerais: 30161-970 Belo Horizonte-MG, Brazil
[2] Google Brasil: Av. Abraão Caram, 430, 4^0 andar - Pampulha, Belo Horizonte-MG, Brazil

CIKM'05, October 31–November 5, 2005, Bremen, Germany.
ACM 1-59593-140-6/05/0010.

1. INTRODUCTION

The huge volume of information now available on the Web has posed challenges to the users. Any short query presents the user with thousands of answers. If the first 10-20 answers are not satisfactory, the user has to sift the answers of his interest among dozens, even hundreds, of answers. Frequently, he gets impatient and takes one of two actions: he rewrites his query in a different form or he simply gives up.

If our user is more persistent, he will rewrite his query trying to make it more specific. It is our believe that, as users of Web search engines become more knowledgeable, they will write more specific, longer, and complex queries (we consider here that a *long* or *complex* query is one that contains two or more terms).

The problem that we face can then be formulated as follows:

> *Given a conjunctive user query, is it reasonable to structure the query in smaller conjunctive components? When should this be attempted? Is it possible to improve precision through one such mechanism?*

Our proposal to solve these questions is a new technique for automatically structuring queries based on *maximal termsets*, a concept directly derived from association rules theory [2], which we will introduce later on. Our technique is referred to as *MAXTERM*, and the key idea is that information derived from the distributions of the query conjunctive components in the document collection can be used to guide the query structuring process. Our approach allows to naturally produce answers to queries that otherwise would lead to empty result sets.

There are several works on structured queries [1, 5, 4]. However, all these approaches suffer from one of the following drawbacks: (i) can not be used for general document collections, or (ii) require a syntactic knowledge base, or (iii) limit the number of terms in term correlations.

2. THE MAXTERM MECHANISM

In this section we present our approach for automatically structuring queries. It is composed by three components: decomposition, selection, and ranking criteria. The decomposition criterion defines how a query is divided into subqueries. The selection criterion defines which subqueries are used to compose the structured query. The ranking criterion defines how the documents that satisfy the subqueries are ranked.

Decomposing Queries into Structured Queries

Considering a conjunctive query formed by n terms a n-termset S, where a n-termset is just a set of n terms. We consider a structured query as a disjunction of conjunctive queries.

Given a query containing n terms, there are $2^n - 2$ subsets that can be derived from it (the power set minus the empty set and the n-termset). In practice, only the subsets that do occur in the text collection are computed. Further, the number of query terms is on average smaller than 10. The combination of those factors allow efficient computation, as we later discuss.

Selecting Structured Queries

In this section we discuss how to select the structured queries that are likely to produce more relevant answers. For that we need the concepts of frequent and maximal termsets.

Definition 1. A termset S_i is a frequent termset if its frequency, i.e., the number of documents containing S_i, is greater than or equal to a given threshold, referred to as minimal frequency.

Definition 2. A maximal termset MS_i is a frequent termset that is not a subset of any other frequent termset.

In terms of the selection criterion for structuring queries, the frequency requirement comes from the intuition that a termset should have a minimal relevance for being considered as a candidate to compose a structured query.

Maximal termsets provide a natural formalism for structuring complex user queries into smaller components that find support in the document collection. This leads to improved results as we verify through experimentation.

Determining maximal termsets is an extension of the data mining problem of mining maximal frequent itemsets. Our maximal termsets enumeration algorithm is based on an efficient algorithm called GENMAX [2]. We adapted GENMAX to handle terms and documents instead of items and transactions, respectively.

Ranking Structured Queries

We use as a basis for ranking structured queries the *set-based vector model* (SBM) [3]. In SBM, weights are associated with the termsets in a document or query, instead of terms. These weights are a function of the number of occurrences of the termset in a document and in the whole collection. In here, we have adapted the original set-based model to use the BM25 weighting scheme. Our approach computes the similarity between a document and the user query as the scalar product between the document vector $\vec{d_j}$, $1 \leq j \leq N$, and the query vector $\vec{q}$, where each vector corresponds to weighted maximal termsets generated for the query terms.

3. EXPERIMENTAL RESULTS

This section describes the experimental results. Our experiments were performed in a Linux-based PC with an AMD-athlon 2600+ 2.0 GHz processor and 512 MBytes of RAM using two reference collections: TREC-8 and WBR-04. The WBR-04 collection is composed of a database of 15,240,881 Brazilian Web pages, under the domain ".br", and a total of 100 example queries.

Retrieval Evaluation

We report results for our approach (MAXTERM), the set-based model (SBM), the probabilistic model using the BM25 weighting scheme (BM25), and the vector space model (VSM) for processing conjunctive queries. We have used the same parameters tuned for SBM in [3] for this work. The BM25 parameters was set with the following values: $k_1 = 1.2$, $k_2 = 0$, $k_3 = 1000$, and $b = 0.75$.

Overall average precision is presented in Table 1. For the TREC-8 test collection, SBM provided a nice gain of 27.44% relative to VSM, and 15.86% relative to BM25 in average precision, while our approach boosted this gain to 36.94% and to 28.03%, respectively. MAXTERM outperforms SBM because it takes into account just the co-occurrence patterns that represent meaningfully "entities" found in the document collections (maximal termsets).

For the WBR-04 test collection, while SBM yields a gain of 23.37% relative to VSM, and 10.90% relative to BM25, our approach leads to a gain of 37.11% and of 23.25%, respectively. That is, our query structuring mechanism is also useful in the context of the Web, and our the gains might represent the difference between a good and a bad answer set for a complex query.

Table 1: Average precision for the evaluated collections.

Collection	Average Precision (%)			
	VSM	*BM25*	*SBM*	*MAXTERM*
TREC-8	22.41	24.65	28.56	30.69
WBR-04	20.18	23.56	26.13	29.04

Performance Evaluation

We compare our approach to SBM, to BM25, and to VSM, when query response times are considered. We observe that SBM takes execution times 14.24% and 42.62% larger than the VSM, 12.65% and 39.86% larger than BM25 for the WBR-04 and the TREC-8 collections, respectively. The execution time increase for MAXTERM ranges from 9.94 to 20.97% relative to VSM, 8.41 to 18.62% relative to BM25 for the WBR-04 and the TREC-8 collections, respectively. The results show that MAXTERM outperforms SBM considering both retrieval effectiveness and execution time.

4. CONCLUSIONS AND FUTURE WORK

We presented MAXTERM, a formalism for automatically structuring a user query into a disjunction of smaller conjunctive subqueries. Our approach analyses the document collection and determines the best conjunctive components based on their support in the document collection. We showed that MAXTERM allows significant improvements in retrieval effectiveness, with processing times close to the times to process the vector space model and the probabilistic model.

Computation of termsets might be restricted by proximity information [3]. This is useful because proximate termsets carry more semantic information than standard termsets. For future work we will investigate the behavior of the MAXTERM approach, when proximity information is taken into account.

5. REFERENCES

[1] P. Das-Gupta. Boolean interpretation of conjunctions for document retrieval. In *Journal of the American Society for Information Science*, volume 38, pages 349-368, 1987.

[2] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163-170, 2001.

[3] B. Pôssas, N. Ziviani, B. Ribeiro-Neto, and W. Meira Jr. Set-based vector model: An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems*. To appear.

[4] M. E. Smith. *Aspects of the P-Norm Model of Information Retrieval: Syntatic Query Generation, Efficiency and Theoretical Properties*. PhD thesis, Computer Science Department, Cornell University, 1990.

[5] M. Srikanth and R. Srihari. Exploiting syntactic structure of queries in a language modeling approach to ir. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 476–483. ACM Press, 2003.