# **Enriching Peer-to-Peer File Descriptors using Association Rules on Query Logs**

Nazli Goharian<sup>1</sup>, Ophir Frieder<sup>1</sup>, Wai Gen Yee<sup>2</sup>, Jay Mundrawala<sup>2</sup>

<sup>1</sup> Department of Computer Science, Georgetown University <sup>2</sup> Information Retrieval Laboratory, Illinois Institute of Technology {nazli, ophir}@cs.georgetown.edu, {waigen, mundra}@ir.iit.edu

**Abstract.** We describe a P2P association rule mining descriptor enrichment approach that statistically significantly increases accuracy by greater than 15% over the non-enriched baseline. Unlike the state-of-the-art enrichment approach however, the proposed solution does not introduce additional network load.

### 1 Introduction

Peer-to-peer file sharing is a major Internet application, consuming an estimated 65% of the 2008 United States Internet traffic [1]. Such a major application requires accurate and efficient query processing capabilities.

In P2P file sharing, peers are both clients issuing search queries and servers responding to them. Files, typically audio or video in nature, are replicated across the network. Each copy of a file, namely a replica, is described by a user-provided textual descriptor of limited length. Files are searched for by comparing a query to a file descriptor. Unfortunately, file descriptors often are sparsely defined and result in few query matches. Thus, search accuracy is poor. Frustrated users ultimately issue additional queries, typically to no avail, and introduce unnecessary network traffic.

Poor search accuracy is due in part to the conjunctive query processing paradigm used in P2P file sharing. A relevant replica only matches a query if all query terms are in its descriptor. Given sparsely specified file descriptors and relatively long queries, query to file descriptor mismatch is likely.

To address the sparse descriptor problem, we propose to enrich each peer's replica descriptors with correlated terms. We do so using association rule mining. Descriptor enrichment increases the ability of a query to match relevant replicas, increasing overall retrieval accuracy. Our retrieval accuracy is comparable to that delivered using the state of the art enrichment method [2]. However, our technique is superior to [2] in that it can be performed by individual peers without coordination of others. The previous technique requires peers to share metadata, making them more prone to failures based on malicious nodes or connectivity patterns. Furthermore, they incur additional network traffic, which may reduce system performance. Our technique avoids these hazards.

## 2 Descriptor Enrichment

Each node maintains a log of queries. These queries are logged as the node routes queries, a process in which every node is assumed to participate. The node uses the contents of these logs to derive term correlations. Correlated terms are added to the terms already in the descriptors of the local replicas. These terms are added until a maximum size is reached. Introducing additional terms requires removing the least frequently used term.

We use the *Apriori Association Rule* algorithm to derive correlated terms (although other techniques are possible [3]). Terms that co-occur with a minimum *support* and *confidence* are identified. Support for a term set  $\{t_i, t_j\}$  is defined as the ratio of queries in the query log that contain the term set. The confidence of the rule  $t_i \rightarrow t_j$  indicates the ratio of the queries that contain  $t_i$  also containing  $t_j$ . Support and confidence are formally defined below, where  $\sigma$  is the counting operator and N is the total number of queries in a peer's query log.

Support 
$$(t_i \to t_j) = \frac{\sigma(t_i \cup t_j)}{N}$$
 Confidence  $(t_i \to t_j) = \frac{\sigma(t_i \cup t_j)}{\sigma(t_i)}$ 

Using these discovered term correlations, each peer updates its replicas' descriptors to include the related terms.

# 3 Experimentation

To evaluate our proposed approach, we developed an experimental platform and assigned parameters matching those in the literature [2]. A set of 1,080 unique Web TREC 2GB (WT2G) documents are grouped into 37 interest categories, derived from the Web domains. Three to five categories are assigned to each 1,000 peers based on a Zipf distribution. The documents for a category are similarly distributed using a Zipf distribution. To each peer, 10 to 30 replicas are assigned at initialization based on the category assignments. The descriptor of each replica is initialized with 3 to 10 terms from the original document according to the term distribution within the document. Table 1 summarizes the experimental data and framework.

Table 1. Statistics

Table 1. Statistics			
Peers	1,000		
Categories	37		
Documents	1,080		
Queries	10,000		
Descriptor size (terms)	20		
Initial descriptors size	3-10		
Categories per peer	3-5		
Files per peer at initialization	10-30		
Trials per experiment	10		

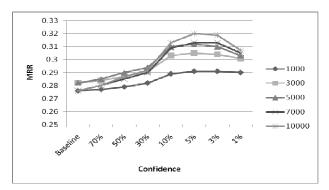


Figure 1. MRR versus confidence and query log size.

Queries are generated by each peer based on its interests. The terms used in each query are based on the term distribution of the corresponding Web TREC documents. The query lengths are based on distributions shown in prior work [2]: 28%, 30%, 18%, 14%, 4%, 3%, 2%, 1% of queries have 1, 2, 3, 4, 5, 6, 7, 8 terms, respectively.

We apply the Apriori algorithm on each peer's query log to *discover* the correlations among terms. The correlations that meet the confidence thresholds are applied to the local replica descriptors to enrich them accordingly by adding the correlated terms to the existing file descriptor terms. We also varied the support threshold for our rules. A support of 0.3% yielded the best results. We omitted support results due to space constraints. Each peer creates a query log from the queries routed through it. Query logs range in size from 1,000 to 10,000 queries.

Accuracy was measured using Mean Reciprocal Rank (MRR):

$$MRR = \frac{1}{N_q} \sum_{i=1}^{N_q} \frac{1}{rank_i}$$

where  $N_q$  is the number of queries issued, and  $rank_i$  is the rank of the desired result in query *i*'s result set. If the desired result is not in the result set, then  $1/rank_i = 0$ .

In each trial, we recorded the MRR of 10,000 queries issued from random peers with and without enriched descriptors. We ran 10 trials and report the average MRR. We compute significance of the results using a paired T-test with a resulting statistical significance of greater than 99%.

## 4. Results

As shown in Figure 1, for all query log sizes, as confidence decreases, MRR increases up to a point (5% confidence) and eventually declines. The increase in MRR stems from the larger descriptors, and hence, higher probability of a match. On the other hand, too low of a confidence threshold results in the *over*-enrichment of the descriptor. Hence, too many descriptors match the query, including those of irrelevant results, which lowers MRR scores.

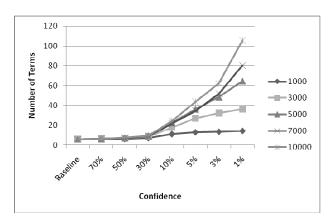


Figure 2. Average descriptor size versus confidence and query log size.

If confidence is set to 5%, the increase in MRR ranges from 5.5% (for a 1,000-query query log) to 16% (for a 10,000-query query log) over the baseline. The larger query log allows greater opportunities for deriving accurate term correlations.

A cost measure for descriptor enrichment is descriptor size. With low confidence and large query logs, cost can be significant: descriptor size may increase from 6 terms to over 100 as shown in Figure 2. The steepest increases in descriptor size occur with confidence values below 5%. Fortunately, with 5% confidence, we yield the best MRR for all query log sizes and the descriptor size is a manageable 42 terms. Therefore, we recommend a confidence of 5%.

### 5 Conclusion

P2P file sharing search accuracy is limited due in part to conjunctive query processing strategy and the relative shortness of the file descriptors. By enriching file descriptors with correlated terms discovered by applying association rules on query logs, we improve accuracy by up to 15%. The benefit of this strategy over previous work is that it does not require additional network traffic to discover the correlations.

The described approach improves MRR performance by greater than 15% when a query log of 10,000 queries and a confidence of 5% are used for rule derivation. The cost at this point is manageable as well: 42 descriptor terms instead of 6. We claim that this is a small price to pay for query accuracy.

## 6 References

- 1. ipoque, ipoque Internet Study, 2008. http://www.ipoque.com/resources/internet\_studies
- 2. Jia, D., et al.: Distributed, Automatic File Description Tuning in P2P File-Sharing Systems, Peer-to-Peer Networking and Applications, 1(2), September 2008.
- J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2006.