# Estimating Global Statistics for Unstructured P2P Search in the Presence of Adversarial Peers

Sami Richardson
Dept. of Computer Science
University College London
Gower St., London WC1E 6BT, UK
sami.richardson.10@ucl.ac.uk

Ingemar J. Cox
Dept. of Computer Science
University College London
Gower St., London WC1E 6BT, UK
i.cox@ucl.ac.uk

## ABSTRACT

A common problem in unstructured peer-to-peer (P2P) information retrieval is the need to compute global statistics of the full collection, when only a small subset of the collection is visible to a peer. Without accurate estimates of these statistics, the effectiveness of modern retrieval models can be reduced. We show that for the case of a probably approximately correct P2P architecture, and using either the BM25 retrieval model or a language model with Dirichlet smoothing, very close approximations of the required global statistics can be estimated with very little overhead and a small extension to the protocol. However, through theoretical modeling and simulations we demonstrate this technique also greatly increases the ability for adversarial peers to manipulate search results. We show an adversary controlling fewer than 10% of peers can censor or increase the rank of documents, or disrupt overall search results. As a defense, we propose a simple modification to the extension, and show global statistics estimation is viable even when up to 40% of peers are adversarial.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*

## Keywords

P2P IR; adversarial IR

## 1. INTRODUCTION

Full-text search across peer-to-peer (P2P) networks has received considerable interest in recent years. P2P architectures can be classified as structured, where content is placed according to defined rules to allow for efficient retrieval, and unstructured, where there are no such rules. To guarantee finding content in an unstructured P2P network it is necessary to search all nodes. Communication costs typically make this infeasible, so search is probabilistic. The Probably Approximately Correct framework [10] was proposed to model the problem of probabilistic search in an unstructured distributed network. The PAC framework assumes that (i) nodes operate independently, (ii) each node indexes a subset of documents from the collection, (iii) the documents indexed are not disjoint across nodes, i.e. each document may be indexed on more than one node, and (iv) a query is performed by sampling a random subset of nodes and combining the results. The *accuracy* of a query is defined as the size of the intersection of the set of documents retrieved by a constrained, probabilistic search and the set that would have been retrieved by an exhaustive search, normalized by the size of the latter.

A PAC architecture gracefully handles the churn associated with nodes joining and leaving the network. This is because the addition of any node compensates for the loss of any other. P2P networks comprised of volunteer nodes typically experience high levels of churn [18], and therefore may be a good match for a PAC architecture. One example is a PAC P2P web search engine, as proposed by Asthana et al [1]. They demonstrated, from a communications bandwidth perspective, the feasibility of using a PAC architecture to store an index of the world wide web on one million volunteer nodes, and to handle a query load equivalent to that seen by the Google web search engine.

However, in a PAC architecture each node is only aware of the documents it indexes, and typically does not have access to the global statistics of the entire document collection that are often used by modern information retrieval models. Without these statistics, a node may not be able to correctly score and rank its documents when responding to a query. As a consequence, the accuracy of queries may not reach the level predicted by the PAC framework. In this paper we evaluate a solution that requires only a small modification to the PAC query procedure. When each node involved in a query returns a list of matching documents, we propose that it also returns information on statistics derived from its local index. After responses from all nodes have been received, this information is used to calculate an improved estimate of global statistics, and the retrieved documents are then scored and ranked again to form a new, potentially more accurate top-$k$ result list. We test this technique with two examples of modern retrieval models, BM25 [17], and a language model with Dirichlet smoothing [22]. We show that accuracy approaches the theoretical value predicted by


*SIGIR'14,* July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609567.

the PAC model, thereby overcoming a previous limitation of the architecture.

It should be expected that a PAC P2P network comprised of volunteer nodes will be subject to malicious behavior. When a secure method is used to select the random set of nodes for each query, such as a gossip-based secure peer sampling service like Brahms [4], the random nature of a PAC architecture makes it relatively resilient to attack. Unfortunately, the global statistics estimation technique greatly increases vulnerability. We demonstrate this by first identifying how an adversary can introduce malicious nodes to perform three attacks: (i) censorship of a document, (ii) promotion (increasing the rank) of a document, and (iii) disruption of overall search results. We then develop theoretical models of these attacks, assuming the global statistics estimation technique is not used. This establishes a baseline of vulnerability. Next, we outline how an adversary can corrupt the global statistics estimation technique, and through simulations demonstrate the potential for manipulation of search results is much greater than for the PAC architecture baseline.

As a defense, we propose that the querying node measures the *skewness* of global statistics information returned from nodes, and filters out values that appear to be manipulated. We show the technique to be highly effective, withstanding up to 40% of malicious nodes before query results are significantly affected.

## 1.1 Paper Overview

In Sect. 2 we discuss related work. In Sect. 3 we review the PAC framework and provide details of BM25 and the language model. In Sect. 4 we modify the PAC query procedure to incorporate the estimation of global statistics, and evaluate its effectiveness. In Sect. 5 we investigate how this increases vulnerability to attack, and in Sect. 6 we propose a defense. Finally, in Sect. 7 overall conclusions are drawn.

# 2. RELATED WORK

The problem of estimating global statistics for P2P information retrieval (P2P IR) is similar to that of estimating corpus statistics for distributed information retrieval (DIR). The multi-database model of DIR assumes that (i) a query is sent to a subset of the most promising databases, (ii) each database returns matching documents, and (iii) results from each database are merged into a final ranked result list [5]. A PAC P2P architecture can be thought of as a special case of this model, where queries are sent to a random subset of databases (nodes), and where each database uses the same retrieval algorithm and contains documents randomly selected from the same central document collection.

In DIR, document scores assigned by different databases may be based on different corpus statistics and retrieval algorithms, and therefore may not be directly comparable. To correctly merge results from different databases, scores from each database can be normalized. When databases are *uncooperative*, and do not aid in this task, normalized scores can be estimated from a sample of documents obtained by submitting queries [6], but normalization is easier when databases are *cooperative* and share local database information, such as corpus statistics. Viles et al. proposed that databases periodically share corpus statistics, so that all databases use the same corpus statistics [19]. However, this may be impractical when there are a very large number of databases (or equivalently, nodes in a P2P network). Callan et al. suggested that corpus statistics be requested from databases before each query, and then passed along with the query [7]. All databases responding to the query can then use the same corpus statistics. However, the increase in query latency may be unacceptable. Kirsch et al. proposed that each queried database returns local corpus statistics, in addition to the result list [15]. The corpus statistics from all databases are then combined, and new, normalized scores are calculated for each returned document. This is similar to the technique we use in this paper. Our work differs, because instead of using a deterministic architecture, we specifically consider a PAC P2P architecture, where documents are randomly replicated across peers, and queries are directed to a random subset of peers.

P2P IR differs from DIR in that P2P networks are typically intended to scale to a much larger number of nodes, potentially thousands or tens of thousands, and are characterized by much higher levels of churn. A number of solutions have been proposed to overcome the lack of global statistics at each node in P2P networks. PlanetP [11] is a P2P information retrieval system that efficiently routes queries to nodes containing relevant documents by using a compact summary of the entire document index maintained at each node. The document frequency global statistic, which is the proportion of documents that index a given term, is not available at each node, so it is difficult to rank documents with the commonly used measure of term frequency-inverse document frequency (TF-IDF). However, the summary index at each node makes it possible to determine peer frequency, the proportion of peers that index at least one document with a given term, and this is used to calculate the measure of term frequency-inverse peer frequency (TF-IPF). The performance of TF-IPF is shown to be similar to that of TF-IDF. Unfortunately, for many other P2P architectures, including PAC, it is not substantially easier to calculate peer frequency than it is to calculate document frequency.

Lu et al. considered search of text based libraries in hierarchical P2P networks [16]. They assume some nodes act as top-level 'hubs' and provide a directory service for low-level 'leaf' nodes that contain text libraries. A query is routed to one or more hub nodes, which in turn route it to appropriate leaf nodes or pass it on to be handled by other hub nodes. The responses from leaf nodes are returned down the query path. Hub nodes maintain global statistics for all connected leaf nodes, and also share these global statistics with other hub nodes. This allows hub nodes to normalize document scores in query responses and merge them into a ranked list before passing the query response back down the query path. As a result, the user is provided with a correctly merged ranked result list. This can be a very effective solution, but it is only applicable to hierarchical P2P architectures.

Chen et al. proposed a hybrid structured/unstructured P2P system for full text-search [9]. The structured component efficiently handles multi-term queries, while the unstructured component gathers global statistics at each node using a gossip protocol. This allows each node to maintain up-to-date global statistics, but comes at the cost of extra inter-node communication traffic.

Witschel et al. [21] showed that reasonable estimates of global statistics can be derived by requesting statistics from random nodes. Our approach also amounts to receiving statistics from random nodes, but takes advantage of the

mechanism already in place to perform queries, whereas in theirs the random sampling is implemented alongside the mechanism to perform queries. Witschel et al. also showed the effectiveness of random sampling can be improved by combining it with a small reference corpus of global statistics on each node. However, this may be less effective with a dynamically changing document collection, and is unnecessary with our approach because global statistics are derived from a large proportion of the document collection.

A major part of our contribution is an analysis of the adversarial manipulation of global statistics. While the impact of malicious nodes in P2P networks has been widely studied [20], we are unaware of investigations into attacks against global statistics. Bender et al. [3] reduce bias of estimates of document frequency using *hash sketches*. Each node creates a hash sketch to provide a compact synopsis of documents that contain a query term, and hash sketches are combined to calculate document frequency. However, this is intended to reduce bias arising from the overlap of document collections across nodes, and not bias caused by adversarial nodes.

# 3. PRELIMINARIES

We first review the fundamental concepts of the PAC framework [10]. This earlier work assumes there is no adversarial behavior. In Sect. 5 we drop this assumption. Let there be $n$ homogenous nodes in the network, $m$ unique documents in the collection, and each node indexes $\rho$ documents. Let the total index capacity of the network be $R$. There are $r_i$ copies of each document $d_i$ replicated across the indexes of nodes, such that $\sum_i r_i = R$. Documents are uniformly randomly replicated, so $r_i = \frac{R}{m}$. Queries are sent to $z$ randomly selected nodes, and relevant documents are combined and ranked to form a top-$k$ result list. The probability of finding $c$ copies of a document $d_i$ is binomially distributed. It was shown [10] that the probability $P(d_i)$ of finding at least one copy of document $d_i$ is given by

$$P(d_i) = 1 - \left(1 - \frac{\rho}{m}\right)^z \ . \tag{1}$$

For (1) to hold, the number of documents returned from each queried node, $k'$, needs to be greater than or equal to $k$. In [10] this was implicitly assumed. In this paper, since we vary $k'$, we explicitly state this requirement.

Using the property of exponential functions, (1) can be approximated with

$$P(d_i) \approx 1 - e^{-\frac{z\rho}{m}} \ . \tag{2}$$

As a consequence, for a fixed collection size $m$, the probability of finding $d_i$ is determined by the product $z\rho$.

In information retrieval, typically there are multiple documents that are relevant to a query. Let $\mathcal{D}_k(j)$ be the *global* top-$k$, the set of top-$k$ documents retrieved for query $j$ from the full document collection, and $\mathcal{D'}_k(j)$ be the *local* top-$k$, the set retrieved from a constrained search of $z$ nodes. The accuracy $a_j$ for query $j$ is then defined [10] as

$$a_j = \frac{|\mathcal{D}_k(j) \cap \mathcal{D'}_k(j)|}{|\mathcal{D}_k(j)|} \ . \tag{3}$$

It was shown [10] that each query is expected to retrieve $k \cdot P(d_i)$ documents out of the global top-$k$, and therefore expected average accuracy is given by

$$E(a_j) = \frac{k \cdot P(d_i)}{k} = P(d_i) \ . \tag{4}$$

## 3.1 BM25 Ranking Function

The first ranking function we evaluate the global statistics estimation technique for is BM25. Let $\mathcal{C}$ be the set of documents in the collection and $T$ be the set of terms in a query. The score, $s_{BM25}(d, T)$, assigned to document $d$ for query $T$ is then given [17] by

$$s_{BM25}(d, T) = \sum_{t \in T} w(t) \cdot s(t, d) \ , \tag{5}$$

where

$$s(t, d) = \frac{TF(t, d) \cdot (k_1 + 1)}{TF(t, d) + k_1 \left(1 - b + b \cdot \frac{DL(d)}{AVGDL}\right)} \ , \tag{6}$$

$w(t) = \log \frac{1}{P_{doc}(t)}$, $k_1$ and $b$ are free parameters, $TF(t, d)$ is the term frequency of term $t$ in document $d$, $DL(d)$ is the number of terms in document $d$, i.e. its length, and $AVGDL$ is the average document length across all documents in $\mathcal{C}$. $P_{doc}(t)$ is the probability of a document in collection $\mathcal{C}$ containing term $t$, and is calculated with

$$P_{doc}(t) = \frac{DF(t, \mathcal{C})}{|\mathcal{C}|} \ , \tag{7}$$

where $DF(t, \mathcal{C})$ is document frequency, the number of documents from collection $\mathcal{C}$ that contain the term $t$.

Each node has access to or can calculate all the parameters of (5), with the exception of $P_{doc}(t)$ and $AVGDL$. These are the global statistics for the collection, which we need to estimate.

## 3.2 Language Model with Dirichlet Smoothing

The second ranking function we consider is a language model with Dirichlet smoothing. For a language model, the score, $s_{lang}(d, T)$, assigned to each document $d$ for query $T$ is given by

$$s_{lang}(d, T) = \prod_{t \in T} p(t|d) \ , \tag{8}$$

where $p(t|d)$ is the probability of the language model of document $d$ generating term $t$, and is given by

$$p(t|d) = \frac{TF(t, d)}{DL(d)} \ . \tag{9}$$

This does not require global statistics of the collection. However, to prevent a score of zero if a query term is not present in a document, it is common to use *smoothing*. Various techniques have been proposed [22] that assign a non-zero value to $p(t|d)$ if the term is missing. In this paper we consider Dirichlet smoothing, for which $p(t|d)$ is given by

$$p(t|d) = \frac{TF(t, d) + \mu \cdot P_{coll}(t)}{DL(d) + \mu} \ , \tag{10}$$

where $\mu$ is a free parameter to control the amount of smoothing, and $P_{coll}(t)$ is the probability of term $t$ being generated from the collection. $P_{coll}(t)$ is given by

$$P_{coll}(t) = \frac{\sum_{d \in \mathcal{C}} TF(t, d)}{\sum_{d \in \mathcal{C}} DL(d)} \ , \tag{11}$$

and is the global statistic we need to estimate.

# 4. GLOBAL STATISTICS ESTIMATION

We now outline a modification to the PAC query procedure that allows the estimation of global statistics. Each node, $u$, uses its local collection of documents, $L_u$, to calculate an initial estimate of the retrieval model global statistics. Using these estimated statistics the node calculates the retrieval model score for each term from all documents in $L_u$. These scores are then added to an index for use when scoring documents for queries. The query procedure is as follows.

1. A querying node issues a query comprised of a set of terms, $T$, to $z$ random nodes (including itself).
2. Each queried node, $u$, then:
   (a) Compiles a top-$k'$ result list of the highest ranked documents from $L_u$ for query $T$, using the previously calculated scores.
   (b) The node returns to the querying node two sets of information: $R_u$ and $G_u$. The former contains summary information for the top-$k'$ documents that the querying node needs to produce a final top-$k$ result list (e.g. document id, parameters of the document required to calculate its score). The latter contains information to estimate the document collection global statistics, used by the retrieval model scoring algorithm.
3. On receiving responses from all $z$ queried nodes, the querying node:
   (a) Calculates new, improved estimates of global statistics based on $G_u$ returned from each node.
   (b) Calculates a score for each received document using summary information from $R_u$ and the new global statistics.
   (c) Ranks documents by their new score, and presents a final top-$k$ result list to the user.

For BM25, the estimate of the global statistic $P_{doc}(t)$ is calculated in Step 3(a) with

$$\hat{P}_{doc}(t) = \frac{\sum_{u \in Z} DF(t, L_u)}{\sum_{u \in Z} |L_u|} , \quad (12)$$

and the estimate of the global statistic $AVGDL$ is calculated with

$$\hat{AVGDL} = \frac{\sum_{u \in Z} \sum_{d \in L_u} DL(d)}{\sum_{u \in Z} |L_u|} . \quad (13)$$

Therefore, for BM25, $G_u$ consists of $DF(t, L_u)$ for $t \in T$, $|L_u|$, and $\sum_{d \in L_u} DL(d)$.

For the language model, the estimate of the global statistic $P_{coll}(t)$ is calculated in Step 3(a) with

$$\hat{P}_{coll}(t) = \frac{\sum_{u \in Z} \sum_{d \in L_u} TF(t, d)}{\sum_{u \in Z} \sum_{d \in L_u} DL(d)} , \quad (14)$$

requiring $G_u$ to consist of $\sum_{d \in L_u} TF(t, d)$ for $t \in T$, and $\sum_{d \in L_u} DL(d)$.

For both BM25 and the language model, the summary information $R_u$ consists of $TF(t, d)$ for $t \in T$ and $DL(d)$, for each document $d$ in the top-$k'$ result list.

The collections $L_u$ on each node used by (12) to (14) are not disjoint across nodes, but because a PAC architecture distributes documents uniformly randomly, global statistics are, on average, unaffected.

## 4.1 Evaluation

We begin our evaluation of the above technique by first assuming there are no malicious nodes present. This demonstrates the maximum gain in query accuracy. In Sect. 5 we then consider the risk that malicious nodes may be able to manipulate search results by returning corrupt global statistics information.

### 4.1.1 Experimental Setup

A simulated network of $n = 10,000$ nodes was used. The document collection, $\mathcal{C}$, was comprised of $m = 1,692,096$ documents from the WT10g [2] web corpus. Documents were uniformly randomly distributed across nodes so that each node indexed $\rho$ documents. Fifty queries were drawn randomly from the TREC 2009 Million Query track [8] and used as the query test set. Each query was performed using the technique described above, where each queried node returned the top $k' = 10$ matching documents, and the accuracy of the final top-10 list calculated with (3). Each query was repeated for a total of ten repetitions, and the average accuracy across queries for a given value of $z$ recorded. In our simulations we chose parameter $z$, the number of nodes a query is issued to, and $\rho$, the number of documents indexed per node, such that the theoretical expected average accuracy given by (4) would be 0.9. For values of $z = 1, 2000, 4000, 6000, 8000, 10000$ this meant corresponding values of $\rho = 1692096, 1946, 973, 649, 486, 389$. Such large values of $z$, and correspondingly small values of $\rho$, were chosen so that the global statistics technique was evaluated under the most challenging circumstances. These experiments were then repeated, first with the querying node using only its local documents to estimate collection global statistics in step (3a), and then again assuming each node had access to the global statistics of the document collection.

We performed the above experiments for both BM25 and the language model. For the former, the free parameters were $k_1 = 2.0, b = 0.75$, and for the latter $\mu = AVGDL$. These are typical choices [17].

### 4.1.2 Results

Figure 1(a) shows average accuracy of queries for different values of $z$, for BM25. There are curves for different combinations of the global statistics $P_{doc}(t)$ and $AVGDL$, derived either from assumed knowledge of the whole collection (*coll*), or from only the collection on the querying node (*node*). As would be expected, when both $P_{doc}(t)$ and $AVGDL$ were derived from the entire collection, average accuracy was about 0.9 for all values of $z$. However, when $P_{doc}(t)$ or $AVGDL$ were estimated only from the index of the querying node, accuracy in general decreased as $z$ increased, i.e. as the number of documents per node, $\rho$, decreased and thus the number of documents from which global statistics could be estimated decreased. Deriving $P_{doc}(t)$ from only the index of the querying node caused a drop in accuracy of up to nearly 35%, whereas doing the same for AVGDL caused a less severe drop of up to about 10%. Figure 1(b) shows the results for the language model. Here the estimated global statistic is $P_{coll}(t)$, and using documents only from the querying node to derive the estimate resulted in a drop of up to about 20%.

Fig. 1(c),(d) show the results when the global statistics estimation technique is used. There are curves for different values of $k'$, i.e. the maximum number of results returned from each of the $z$ queried nodes. For BM25, the global

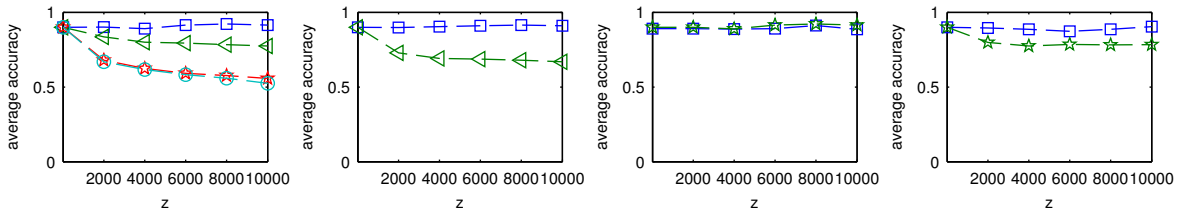


**Figure 1: Average accuracy of queries, where each pair of $z, \rho$ values is chosen to achieve a theoretical expected average accuracy of 0.9. Global statistics are estimated from either the whole collection (*coll*) or from just the querying node (*node*). (a) is for BM25, where the curves from top to bottom are for (coll $P_{doc}(t)$, coll $AVGDL$), (coll $P_{doc}(t)$, node $AVGDL$), (node $P_{doc}(t)$, coll $AVGDL$), and (node $P_{doc}(t)$, node $AVGDL$). Note the last two curves overlap. (b) is for the language model, where the curves from top to bottom are for (coll $P_{coll}(t)$), and (node $P_{coll}(t)$). (c) and (d) are for BM25 and the language model respectively, using the global statistics estimation technique with $k' = \rho$ (top) or $k' = 10$ (bottom). Note the two curves in (c) overlap.**

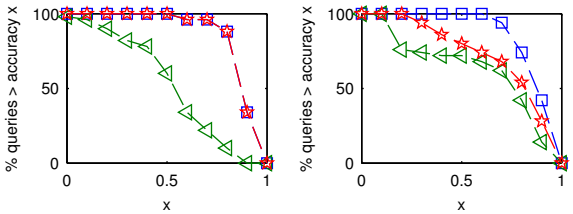


**Figure 2: The percentage of queries that achieve accuracy $x$ when global statistics are estimated from either the whole collection (*coll*) or from just the querying node (*node*). (a) is for BM25, where the curves from right to left are for (coll $P_{doc}(t)$, coll $AVGDL$), ($\hat{P}_{doc}(t)$, $A\hat{V}GDL$), and (node $P_{doc}(t)$, node $AVGDL$). Note the first two curves overlap. (b) is for the language model, where the curves from right to left are for (coll $P_{coll}(t)$), ($\hat{P}_{coll}(t)$), and (node $P_{coll}(t)$). For $z = 10,000$, $\rho = 389$, and $k' = 10$.**

statistics estimation technique achieves accuracy that is very close to the theoretical value of 0.9, for all values of z. The same is true for the language model for $k' = \rho$, but for $k' = 10$ average accuracy drops to 0.8 for larger values of $z$, which is about 10% lower than the theoretical value.

We are also interested in how accuracy varies across different queries. For parameters $z = 10,000$, $\rho = 389$, and $k' = 10$, Fig. 2(a),(b) show the proportion of queries that achieve a given accuracy, for BM25 and the language model respectively. With only 389 documents per node these results represent performance under challenging circumstances. Nevertheless, for BM25, the proportion of queries achieving a given accuracy when using the global statistics estimation technique *($\hat{P}_{doc}(t)$, $A\hat{V}GDL$)* is almost identical to the case where collection global statistics are available at each node *(coll $P_{doc}(t)$, coll $AVGDL$)*, e.g. about 95% of queries achieve an accuracy of at least 0.7. When global statistics are derived only from documents at the querying node *(node $P_{doc}(t)$, node $AVGDL$)*, this figure falls to 15%. For the language model, the global statistics estimation technique does not perform quite as well, with about 65% of queries achieving an accuracy of at least 0.7, compared to about 90% for when global statistics are available at each node, and 60% when global statistics are estimated from only documents at the querying node. However, for the global statistics technique over 95% of queries achieve an accuracy of at least 0.3, compared to less than 80% when global statistics are estimated from only documents at the querying node.

### 4.1.3 *Discussion*

The experiments showed that the global statistics estimation technique can achieve an average query accuracy that is very close to what would be attained if global statistics had been available at each node, at least for larger values of $k'$, even for extreme cases where each node indexes only a small proportion of the document collection. To understand why this is the case, we observe that for each query global statistics are estimated from $z\rho$ documents, a potentially very large sample. Of course these documents are unlikely to be distinct. The number of distinct documents, $n_{\text{distinct}}$, is between $\rho$ and $\min(m, z\rho)$. It is straightforward to show that the expected value of $n_{\text{distinct}}$ is given by

$$E(n_{\text{distinct}}) = P(d_i)m \ , \tag{15}$$

where $P(d_i)$ is given by (1). It follows that the expected *coverage* for an estimate, i.e. the proportion of documents in the collection that the estimate is based on, is given by

$$E(\text{Coverage}) = \frac{n_{\text{distinct}}}{m} = P(d_i) \ . \tag{16}$$

The expected average accuracy for a top-$k$ query, as given by (4), is also equal to $P(d_i)$. Therefore, by choosing network parameters $\rho$ and $z$ to increase theoretical expected average accuracy, coverage is also increased for the global statistics estimates, and accuracy moves towards the upper bound predicted by the PAC framework. For example, a network could be designed to achieve high theoretical accuracy, such

as 0.9, which means that estimates of global statistics will be based on 90% of the document collection, and would be expected to be very close to the correct global statistics.

As was apparent for the language model with $k' = 10$, the effectiveness of the global statistics estimation technique may be reduced when $k' < \rho$. Since the top-$k'$ result lists are calculated using global statistics estimated from just the local index of one node, ranking may be incorrect, and therefore relevant documents may not be returned to the querying node. Consequently, no matter how accurate the final global statistic estimate is, these documents will never appear in the top-$k$ result list presented to the user. However, in practice it is likely that a large value of $k'$ can be used, since the communication cost associated with each result in the result list is small. For example, both BM25 and the language model require only a document id, and values for term frequency and document length to be returned for each result.

# 5. ADVERSARIAL ATTACKS

We now show that if an adversary can introduce malicious nodes, the global statistics estimation technique can be subverted to manipulate search results. In the analysis that follows, it is assumed an adversary controls the proportion $f \in [0, 1]$ of the $n$ nodes in the network. To ensure no node has a greater influence on search results than any other, each node is restricted to indexing the same number of documents, $\rho$. In practical systems the capacity of each node may differ, and can be dealt with by allowing nodes with higher capacities to operate multiple 'virtual' nodes, each of which has capacity $\rho$. This resembles the Sybil attack [12], where an adversary impersonates a large number of nodes to control the network. Therefore, the defensive techniques discussed in [12] to restrict the number of nodes operated by an individual need to be applied to both virtual and physical nodes. For example, a check can be made to verify that only a limited number of virtual or physical nodes are associated with an email address.

We consider the following attacks.

- **Censorship.** Reduce the likelihood of a target document appearing in the final top-$k$ result list.
- **Promotion.** Increase the rank of a target document so that it is more likely to appear in the final top-$k$ result list, and if it does appear, to rank higher.
- **Disruption.** Reduce the 'correctness' of the final top-$k$ result list, i.e reduce accuracy, as given by (3).

A node responding to a query returns $R_u$ for the top-$k'$ matching documents, and $G_u$. The former contains result summary information, such as document id, document length etc, and the latter contains information on global statistics. An adversary can perform the above attacks by using malicious nodes to return corrupt information for $R_u$ and/or $G_u$. (We assume that malicious nodes cannot construct and return corrupt documents that will score highly for a query; this can be enforced by requiring documents to be digitally signed by a trusted third party.)

In our analysis we initially assume the global statistics estimation technique is not used, and attacks only corrupt $R_u$. This establishes the baseline vulnerability inherent to the PAC architecture. We then consider the increase in attack effectiveness that arises when the global statistics estimation technique is used and an adversary can also corrupt $G_u$.

## 5.1 Baseline Vulnerability

Malicious nodes can perform the Censorship attack by returning corrupt summary information for $k$ documents in $R_u$, such that each document would score higher than the target document and prevent it from appearing in the final top-$k$ result list. For the Promotion attack, malicious nodes would always include the target document in $R_u$, along with corrupt summary information so that it will outrank any other. For the Disruption attack, malicious nodes would return $k$ irrelevant documents, all with corrupt summary information that ensures they will outrank other documents. To have an effect on a query, these attacks require only a single malicious node to be one of the $z$ nodes randomly sampled. If the proportion $f$ of nodes are malicious, the probability $P(m_i)$ of a query visiting at least one malicious node is

$$P(m_i) = 1 - (1 - f)^z \quad . \tag{17}$$

For $z = 1,000$, it would require an adversary to control only $f = 0.3\%$ of nodes for there to be a 0.95 probability of a malicious node being visited by the query, and therefore allow the adversary to manipulate on average 95% of queries.

However, incorrect summary information in $R_u$ can be detected by retrieving the documents. For example, a querying node, on receiving responses from all queried nodes, could retrieve the documents in the final top-$k$ result list, calculate the summary information for each, and only display to the user results with correct scores. The extra latency and communication costs involved with this may be unacceptable, so an alternative is to display the top-$k$ result list to the user, unchecked. Only when a user chooses to view a document is it retrieved and the score verified. If the score proves to be incorrect, then the document is not made available to the user. Since incorrect summary information can be easily detected, we assume an adversary does not perform attacks using this approach. A more subtle, and less easily detectable alternative, is for malicious nodes to return correct summary information for documents in $R_u$, but to exclude specific documents. Each node indexes random documents, so it is more difficult to determine if a node is not returning a given document because it is behaving maliciously, or because the document is simply not in its index. Attacks carried out by excluding documents form the baseline of vulnerability for a PAC architecture.

### 5.1.1 Censorship

The Censorship attack can be performed by malicious nodes excluding the target document. Let $P'(d_i)$ be the probability of retrieving document $d_i$ when the proportion $f$ of nodes are malicious and exclude it. From (1) it is straightforward to show that $P'(d_i)$ is given by

$$P'(d_i) = 1 - \left(1 - \frac{\rho}{m}\right)^{z(1-f)} \quad . \tag{18}$$

Using the property of exponential functions, this can be approximated with

$$P'(d_i) \approx 1 - e^{-(1-f)\frac{z\rho}{m}} \quad . \tag{19}$$

Equations (2) and (4) can be used to estimate the expected average accuracy for a query when there are no malicious

nodes present, $E(a_j)$. $P'(d_i)$, for a given proportion $f$ of malicious nodes, and $E(a_j)$ are both determined by $\frac{z\rho}{m}$. Figure 3 shows the effect on $P'(d_i)$ as $f$ is varied. Each curve depicts different choices of $\frac{z\rho}{m}$ to achieve $E(a_j) = 0.3, 0.6, 0.9$. As $E(a_j)$ increases, resilience to censorship also increases. Typically, $\frac{z\rho}{m}$ would be chosen to achieve high average expected accuracy, so resilience to censorship would be high. For example, when $E(a_j) = 0.9$, it would require about 70% of nodes to be malicious to reduce the probability of finding $d_i$ by 50% from 0.9 to 0.45.

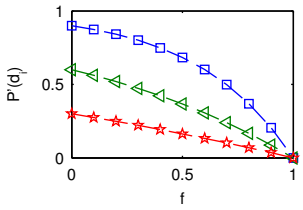


**Figure 3: Probability $P'(d_i)$ of retrieving document $d_i$ when the proportion $f$ of nodes are performing the Censorship attack by excluding $d_i$. For $E(a_j) =$ 0.9 (top), 0.6 (middle), 0.3 (bottom).**

### 5.1.2 Promotion

The Promotion attack can be carried out by censoring documents that rank higher than the target document, thus improving the rank of the target document. If there are $u$ documents in the global top-$k$ for a query that rank higher than the target document, and if malicious nodes never return them when queried, the probability $P(u')$ of retrieving $u'$ documents out of the total $u$ is given by

$$P(u') = \binom{u}{u'} P'(d_i)^{u'} \left(1 - P'(d_i)\right)^{u-u'} , \qquad (20)$$

where $P'(d_i)$ is the probability of retrieving one of the excluded documents, as given by (18). Since this is a standard binomial distribution, the expected number of documents retrieved is

$$E(u') = u \cdot P'(d_i) . \qquad (21)$$

If document $d_i$ is retrieved for a query, then its rank is one plus the number of other documents retrieved that rank higher. Therefore, (21) can be expressed in terms of the expectations of the rank of the target document before the attack, $r_{\text{before}}$, and the rank after, $r_{\text{after}}$:

$$E(r_{\text{after}}) = (E(r_{\text{before}}) - 1)P'(d_i) + 1 . \qquad (22)$$

As for the Censorship attack in Sect. 5.1.1, we consider the effectiveness of this attack for an example system designed to achieve expected average accuracy of 0.9 when no malicious nodes are present. This requires $\frac{z\rho}{m} = 2.3$. An adversary would need to control over $f = 50\%$ of nodes to increase the expected rank of a target document from 10 to 2.

### 5.1.3 Disruption

The Disruption attack can be performed in a similar manner to the Promotion attack, except rather than excluding the $u$ documents that rank higher than a target document, all $k$ documents in the global top-$k$ for a query are excluded. The probability $P(u')$ of retrieving $u'$ documents from the global top-$k$ for the query is given by (20) (where $u = k$), and the expected number of documents retrieved, $E(u')$, is given by (21). If $a'_j$ denotes accuracy for a query $j$ when malicious nodes are censoring all global top-$k$ documents, then $E(a'_j)$ is given by

$$E(a'_j) = \frac{E(u')}{k} = P'(d_i) . \qquad (23)$$

As an example, for $\frac{z\rho}{m} = 2.3$, expected accuracy for queries is 0.9 when no malicious nodes are present. If we assume users find search results acceptable as long as expected accuracy remains above 0.5, then an adversary would need to control over $f = 70\%$ of nodes to reduce expected accuracy below this threshold.

## 5.2 Increased Vulnerability - Global Statistics Estimation

Section 5.1 established a theoretical baseline for the vulnerability of a PAC architecture, which assumes attacks are performed by excluding documents. In this section we investigate the increased vulnerability that the global statistics estimation technique introduces. For the following analysis, the global statistic to be estimated for BM25 is $P_{doc}(t)$, calculated at the querying node with (12), and for the language model $P_{coll}(t)$, calculated at the querying node with (14). BM25 also requires the global statistic AVGDL, but since $AVGDL$ is a single value, it is feasible for every node to store it. For a fixed document collection size, this is trivial to implement; for a collection size that varies, a gossip protocol can be used to compute it [14].

When estimating $P_{doc}(t)$ and $P_{coll}(t)$ with (12) and (14), the numerator and denominator of both equations are values returned from queried nodes. If no limits are placed on these, even a single malicious node can dominate the result. This is prevented for $P_{doc}(t)$, however, since we restrict the capacity of each node to $\rho$. $P_{doc}(t)$ is then estimated with

$$\hat{P}_{doc}(t) = \frac{\sum_{u \in Z} \min\left(\rho, DF(t, L_u)\right)}{\rho \cdot z} . \qquad (24)$$

To prevent a single node dominating the estimate of $P_{coll}(t)$, it is assumed that the sum of document lengths on a node is $AVGDL \cdot \rho$. Approximate estimates of $P_{coll}(t)$ can then be calculated with

$$\hat{P}_{coll}(t) \approx \frac{\sum_{u \in Z} \min\left(\psi, \left(\sum_{d \in L_u} TF(t, d)\right)\right)}{\psi \cdot z} , \qquad (25)$$

where

$$\psi = AVGDL \cdot \rho . \qquad (26)$$

We shall see that this approximation can still yield very good results.

Equations (24) and (25) are more succinctly expressed as

$$\hat{g}_t = \frac{1}{c} \sum_{u \in Z} x_t^{(u)} , \qquad (27)$$

where $\hat{g}_t$ is an estimate of the global statistic $g_t$. For BM25, we have $g_t = P_{doc}(t)$, and $x_t^{(u)}$ equal to the numerator of (24) and $c$ equal to the denominator. For the language model, we have $g_t = P_{coll}(t)$, and $x_t^{(u)}$ equal to the numerator of

(25) and $c$ equal to the denominator. The $G_u$ information returned from each node then consists of $x_t^{(u)} : t \in T$.

We now investigate how an adversary may attempt the attacks from Sect. 5. Experiments were performed for both BM25 and the language model, but since findings for both are similar, for brevity we only present results for BM25.

### 5.2.1 Censorship/Promotion

An adversary can decrease/increase the score of a target document for query $T$ by using malicious nodes to manipulate $\hat{g}_t : t \in T$. However, this will not necessarily decrease/increase the rank, since the scores of other documents may also be decreased/increased. A more effective approach is to iterate through different values of $g_t : t \in T$, calculate the rank of the target document for each, and select the values that minimize/maximize rank. We denote these optimal values as $\dot{g}_t : t \in T$. An adversary then uses malicious nodes to return a corrupt value, $x'_t$, for $x_t^{(u)}$, and manipulates $\hat{g}_t$ to be $\dot{g}_t$. To find the required value of $x'_t$, we observe that during the attack $\hat{g}_t$ can be estimated with

$$\hat{g}_t = \frac{z}{c}\left((1-f)x_t^{(u)} + fx'_t\right) \quad . \tag{28}$$

The value of $x'_t$ is then selected so that $\hat{g}_t = \dot{g}_t$. If the proportion of nodes an adversary controls, $f$, is too small to select a value of $x'_t$ that will satisfy (28), then $\dot{g}_t : t \in T$ are discarded and the iteration repeated until values of $\dot{g}_t : t \in T$ are found that allow (28) to be satisfied.

Figure 4 illustrates the potential effect of these attacks for the queries $T_1$ ='small dog' and $T_2$ ='brown dog' on the rank of two documents selected from the WT10g corpus, $D_1$ and $D_2$. Each curve is calculated with (5) by assuming full access to the document collection, and using global statistic $g_1$ when scoring the first term and $g_2$ for the second. The range of ranks achieved by varying $g_1$ and $g_2$, and therefore the potential for manipulation, is considerable, but depends heavily on the query-document combination.

We simulated these attacks using the experimental setup from Sect. 4.1.1, but with a proportion $f$ of nodes behaving maliciously. Malicious nodes performed the attacks by returning corrupt global statistics, as described above, and by excluding specific documents, as described in Sects. 5.1.1 and 5.1.2. In order to observe the full range of ranks a document may achieve when under attack, the final result list for each query was not restricted to just the top-$k$, i.e. all retrieved documents were treated as important, and each queried node returned results for all documents it indexes, i.e. $k' = \rho$.

Figure 5 shows results for the query $T$ ='small dog', when performing the Censorship attack on $D_2$, and when performing the Promotion attack on $D_1$. There are $z = 2,000$ nodes involved in the query, and each node indexes $\rho = 1,946$ documents. Considering first the Censorship attack (top left), when varying the proportion of malicious nodes from 0 to 10% to 20% to 30%, rank decreases from 5 to 9 to 582 to 2166. Therefore, for top-$k$ queries, when $k = 10$ it would require an adversary to control less than 20% of nodes for the target document to not appear in the final top-10. Compared to the PAC architecture baseline, where correct global statistics are available at each node, and the attack is performed only by excluding $D_2$, then from (18), with 20% of malicious nodes there is still an expected 84% probability of $D_2$ being retrieved and appearing in the final top-10.

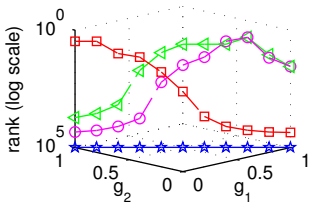


**Figure 4: Effect of global statistics on rank of documents $D_1$ and $D_2$ for queries $T_1$ = 'small dog' and $T_2$ = 'brown dog', where $g_1$ is the global statistic for the first term, and $g_2$ is for the second. For query-document combinations $T_1, D_1$(top left), $T_1, D_2$(top middle), $T_2, D_1$(bottom), $T_2, D_2$(bottom left).**

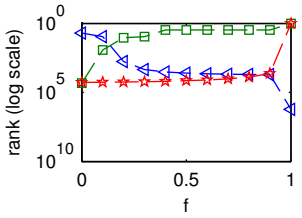


**Figure 5: Censorship attack on $D_2$ by manipulating global statistics (top left). Promotion attack on $D_1$ by manipulating global statistics and excluding documents (top), or just by excluding documents (bottom). For query $T$ = 'small dog', and $z = 2,000$, $\rho = 1,946$.**

From the Promotion attack curve (top), it can be seen that varying the proportion of malicious nodes from 0 to 10% to 20% to 30% increases the rank of $D_1$ from 20778 to 84 to 11 to 9, demonstrating that fewer than 30% of malicious nodes are required to bring $D_1$ into the top-10. The Promotion attack baseline curve (bottom) shows the PAC architecture baseline, which is the theoretical expected rank calculated with (22), when correct global statistics are available at each node, and nodes perform the Promotion attack by only excluding documents. In this case, over 95% of malicious nodes are required to promote $D_1$ into the top-10. Clearly, for both the Promotion and Censorship attacks, the global statistics estimation technique can greatly increase vulnerability to manipulation.

We repeated the simulations with $k = k' = 10$, i.e. only the final top-10 documents were considered important, and each queried node returned the top-10 matching documents. Results were very similar to the previous simulations for documents at ranks 1 to 10.

### 5.2.2 Disruption

An adversary can achieve maximum disruption of a query by using malicious nodes to return responses that make estimates of global statistics at the querying node, $\hat{g}_t : t \in T$, as 'wrong' as possible. For example a global statistic would be assigned a high value, even though it should be low, and vice-versa. The Disruption attack can be represented as an optimization problem, where $\hat{g}_t : t \in T$ are manipulated to maximize the squared difference, $\delta$, between the true global

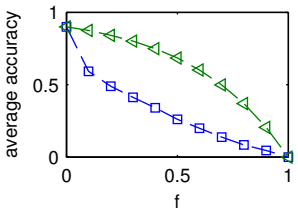


**Figure 6: Disruption attack by manipulating global statistics and excluding documents (bottom), or just by excluding documents (top). For $z = 2,000$, $\rho = 1,946$.**

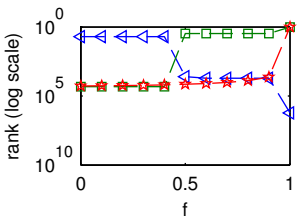


**Figure 7: Censorship and Promotion attacks. As Fig. 5, but with skewness defense in operation.**

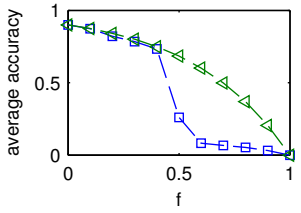


**Figure 8: Disruption attack. As Fig. 6 but with skewness defense in operation.**

statistic $g_t$ and the estimated value $\hat{g}_t$ for each term $t$ in the query, as given by

$$\delta = \sum_{t \in T} (g_t - \hat{g}_t)^2 \ , \tag{29}$$

with $\hat{g}_t : t \in T$ constrained according to the global statistics they represent, e.g. for $g_t = P_{doc}(t)$ or $g_t = P_{coll}(t)$, we require $0 \leq \hat{g}_t \leq 1$. Standard non-linear optimization techniques can be used to find values of $\hat{g}_t : t \in T$ that maximize (29). As for the Censorship/Promotion attacks in Sect. 5.2.1, we denote these optimal values as $\dot{g}_t : t \in T$. Again, an adversary can use malicious nodes to return $x'_t$, so that each value of $\hat{g}_t$, as calculated with (28), becomes $\dot{g}_t$.

Simulations were performed using the setup from Sect. 4.1.1, with fifty randomly selected queries and $k = k' = 10$. Malicious nodes returned corrupt global statistics information, $G_u$, to maximize (29). In addition, they also excluded the global top-$k$ documents for the query, as described in Sect. 5.1.3. Figure 6 shows the results for $z = 2,000$ and $\rho = 1,946$. The bottom curve depicts average accuracy across the fifty queries for different proportions of malicious nodes. Also, as a baseline, the top curve depicts theoretical expected average accuracy, calculated with (23), that assumes correct global statistics are available at each node and that malicious nodes perform the Disruption attack by excluding documents. Clearly, the potential for attack is much greater when the adversary is able to corrupt global statistics. With only 10% of malicious nodes, average accuracy drops from a theoretical baseline of about 0.9 to about 0.6, a nearly 35% fall.

# 6. ROBUST GLOBAL STATISTICS ESTIMATION

Section 5.2 showed that when the global statistics estimation technique is used, even a small proportion of malicious nodes can significantly affect query results. We now propose a defense. The querying node calculates estimates of the global statistics, $\hat{g}_t : t \in T$, using (27) with values of $x_t^{(u)} : t \in T$ returned from each queried node $u$. Values of $x_t^{(u)}$ from non-malicious nodes are expected to be normally distributed, since each node determines the value from its local collection of random documents. If an adversary attempts to bias $\hat{g}_t$ by using malicious nodes to return skewed values of $x_t^{(u)}$, then this normal distribution will be skewed in one direction. We propose that the querying node measures this skewness, $K_t$, using a standard measure [13] given by

$$K_t = \frac{\sqrt{z(z-1)}}{z-2} \left[ \frac{\frac{1}{z} \sum_{u \in Z} (x_t^{(u)} - \bar{x}_t^{(u)})^3}{\left( \frac{1}{z} \sum_{u \in Z} (x_t^{(u)} - \bar{x}_t^{(u)})^2 \right)^{\frac{3}{2}}} \right] \ , \tag{30}$$

where $\bar{x}_t^{(u)} = \frac{1}{z} \sum_{u \in Z} x_t^{(u)}$. If $K_t$ is greater than a threshold $\tau$, the querying node sorts values of $x_t^{(u)}$, and repeatedly discards the largest value until skew is within the limit. Similarly, if $K_t < -\tau$, the smallest value is repeatedly discarded.

We reran the attack simulations from Sect. 5.2, but with the querying node reducing skewness to within the threshold $\tau = \pm 0.1$. Figures 7 and 8 show the results for the Censorship/Promotion, and Disruption attacks respectively. For a proportion of malicious nodes $f < 40\%$, the attacks have very little effect. This is because malicious values of $x_t^{(u)}$ are being removed, and the only impact of the attacks is to reduce the number of non-malicious values available for computation of global statistics. When $f$ rises above 40%, the defense rapidly breaks down due to the proportion of malicious nodes nearing that of the proportion of non-malicious nodes, and therefore it is no longer possible to distinguish between malicious and non-malicious nodes.

The defense is effective because in order to manipulate global statistics, an adversary needs to introduce skew, but the defense directly measures skew and limits it. With this defense the global statistics estimation technique can be safely used to improve query accuracy when fewer than 40% of nodes are malicious. For many situations this may be sufficient. For example, to select a random subset of nodes for each query, a gossip-based secure peer sampling service like Brahms [4] can be used. Brahms can withstand up to 20% of nodes behaving maliciously before sampling becomes significantly biassed. Consequently, it would be Brahms that imposes the limit on the maximum number of malicious nodes tolerated, and not the technique to estimate global statistics.

## 7. CONCLUSIONS AND FUTURE WORK

In unstructured P2P information retrieval, performance can be severely degraded by poor estimates of the global statistics of the collection. For the case of unstructured P2P PAC search, we proposed that a querying node estimates the global statistics of the collection using information derived from the local statistics of the responding nodes. We showed, both theoretically, and experimentally with BM25 and a language model, that such an approach can provide accurate estimates of global statistics and significantly improve retrieval performance. The solution is well suited to a PAC architecture because it requires only a minimal amount of extra information to be returned from queried nodes. Unfortunately, it greatly increases the ability for an adversary to manipulate search results. We identified attacks where an adversary may attempt to (i) censor a document, (ii) promote a document, or (iii) disrupt overall search results. Through theoretical modeling and simulations we showed that while a PAC architecture is resilient to even a large proportion of malicious nodes, when the global statistics estimation technique is used, an adversary would need to control only 10% of nodes to have a significant impact. To protect against this, we proposed that the querying node filters out the most skewed responses, and showed that more than 40% of nodes would need to be malicious before these attacks become effective.

Our work assumed that a peer's local collection consists of a uniform random sample from the global collection. Future work is needed to analyze the case where document sampling is non-uniform, such as when based on document popularity. In this case, we believe that hash sketches, described in Section 2, may form the basis of a solution.

## 8. ACKNOWLEDGEMENTS

Sami Richardson was supported by EPSRC grant no. EP-G037264-1 (Security Science Doctoral Training Centre).

## 9. REFERENCES

[1] H. Asthana, R. Fu, and I. J. Cox. On the feasibility of unstructured peer-to-peer information retrieval. In *Advances in Information Retrieval Theory*, pages 125–138. Springer, 2011.

[2] P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for web retrieval experiments. *Information Processing & Management*, 39(6):853–871, 2003.

[3] M. Bender, S. Michel, P. Triantafillou, and G. Weikum. Global document frequency estimation in peer-to-peer web search. In *Proc. of the 9th Int. Workshop on the web and databases*, 2006.

[4] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, and A. Shraer. Brahms: Byzantine resilient random membership sampling. *Computer Networks*, 53(13):2340–2359, 2009.

[5] J. Callan. Distributed information retrieval. In *Advances in Information Retrieval*, pages 127–150, 2000.

[6] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems (TOIS)*, 19(2):97–130, 2001.

[7] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–28. ACM, 1995.

[8] B. Carterette, V. Pavlu, H. Fang, and E. Kanoulas. Million query track 2009 overview. In *Proceedings of TREC*, volume 9, 2009.

[9] H. Chen, J. Yan, H. Jin, Y. Liu, and L. M. Ni. TSS: Efficient term set search in large peer-to-peer textual collections. *Computers, IEEE Transactions on*, 59(7):969–980, 2010.

[10] I. J. Cox, R. Fu, and L. K. Hansen. Probably approximately correct search. In *Advances in Information Retrieval Theory*, pages 2–16. Springer, 2009.

[11] F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. Planetp: Using gossiping to build content addressable peer-to-peer information sharing communities. In *HPDC'03: Proceedings of the 12th International Symposium on High Performance Distributed Computing, Seattle, WA, USA*, 2003.

[12] J. Douceur. The Sybil attack. *Peer-to-peer Systems*, pages 251–260, 2002.

[13] R. A. Groeneveld and G. Meeden. Measuring skewness and kurtosis. *The Statistician*, pages 391–399, 1984.

[14] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 482–491. IEEE, 2003.

[15] S. T. Kirsch. Document retrieval over networks wherein ranking and relevance scores are computed at the client for multiple database documents, Aug. 19 1997. US Patent 5,659,732.

[16] J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *ECIR'05: Proceedings of the 27th European conference on IR Research, Santiago de Compostela, Spain*, 2005.

[17] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[18] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202. ACM, 2006.

[19] C. L. Viles and J. C. French. Dissemination of collection wide information in a distributed information retrieval system. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 12–20. ACM, 1995.

[20] D. Wallach. A survey of peer-to-peer security issues. *Software Security—Theories and Systems*, pages 253–258, 2003.

[21] H. F. Witschel. Global term weights in distributed environments. *Information Processing & Management*, 44(3):1049–1061, 2008.

[22] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.