Ranked Accuracy and Unstructured Distributed Search

Sami Richardson and Ingemar J. Cox

Dept. of Computer Science, University College London, UK
{sami.richardson.10,i.cox}@ucl.ac.uk

Abstract. Non-uniformly distributing documents in an unstructured peer-to-peer (P2P) network has been shown to improve both the expected search length and search accuracy, where accuracy is defined as the size of the intersection of the documents retrieved by a constrained, probabilistic search and the documents that would have been retrieved by an exhaustive search, normalized by the size of the latter. However neither metric considers the relative ranking of the documents in the retrieved sets. We therefore introduce a new performance metric, rank-accuracy, that is a rank weighted score of the top-k documents retrieved. By replicating documents across nodes based on their retrieval rate (a function of query frequency), and rank, we show that average rank-accuracy can be improved. The practical performance of rank-aware search is demonstrated using a simulated network of 10,000 nodes and queries drawn from a Yahoo! web search log.

Keywords: Unstructured P2P Network, Probabilistic Retrieval.

1 Introduction

Peer-to-peer (P2P) network architectures can be grouped into the categories of structured, unstructured, or a hybrid combination of the two. In this paper we are concerned with the search of unstructured networks, where documents are distributed randomly across nodes. To guarantee finding a document it is necessary to visit every node. In practice, it is usually only feasible to visit a small subset of nodes, and therefore search is probabilistic. Previous work developed a theoretical framework known as probably approximately correct (PAC) search to model this problem [1]. It assumes that (i) nodes operate independently, without communicating with each other, (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.

An advantage of PAC search is that each node can operate autonomously. This means there is very little communication overhead between nodes and the failure of individual nodes has a limited effect on others. A disadvantage is lower performance when compared to deterministic systems using the same resources. This disadvantage forms the basis for measuring a key performance indicator of

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 171–182, 2013.

a PAC information retrieval (IR) system. Specifically, the *accuracy* of a PAC system is defined as the size of the intersection of the documents retrieved by a constrained, probabilistic search and the documents that would have been retrieved by an exhaustive search, normalized by the size of the latter. Other performance indicators include expected search length, bandwidth and storage requirements, which are outside the scope of this paper.

Note that the accuracy is independent of the IR model. This is intentional, as PAC accuracy is intended to only measure the performance degradation caused by not searching over the entire collection. Thus, it is fundamentally different to standard information retrieval metrics in two ways. First, the PAC accuracy does not consider the relevance of documents, but simply measures what fraction of documents will, on average, be retrieved when searching over a random subset of the collection, where it is assumed that the retrieval model is the same for both deterministic and probabilistic search. This assumption may not always be valid, particularly since retrieval models often require statistics of the document collection that may be unavailable to individual peers in a network. In such a case, the PAC accuracy can be considered an upper bound on performance. The second difference is that PAC accuracy ignores the rank order of the retrieved documents. Thus, for example, a PAC system that retrieves the top 50% of relevant documents has the same accuracy as a PAC system that retrieves the bottom 50% of relevant documents. However, from a user-perspective, the latter system would be judged to perform much worse than the former.

The importance of rank order in user perception of IR system performance is well known and a number of performance measures have been proposed, e.g. DCG [2] and RBP [3]. In this paper we modify the definition of PAC accuracy to account for a document's rank, and we refer to this as PAC rank-accuracy. Equations are provided to predict rank-accuracy.

Prior work, discussed in Sect. 2, has shown that the accuracy of P2P search can be significantly improved by replicating documents non-uniformly over the network, based on the popularity of queries, i.e. the query distribution. We extend this work by replicating documents based on their retrieval rate and associated rank. We experimentally demonstrate that such a replication policy significantly improves the rank-accuracy of PAC search.

In Sect. 3 we present our new rank-aware search framework and propose a weighting scheme derived from a rank biased precision metric. The theoretical performance of rank-aware PAC search is evaluated in Sect. 4, and in Sect. 5 experimental results from simulations are presented. Finally, in Sect. 6 conclusions are drawn on the value of rank-aware PAC search and areas for future work are identified.

2 Related Work

Evaluation of P2P IR systems typically focuses both on information retrieval (IR) performance, as well as system performance measured by such factors as communication bandwidth and latency [4,5]. These latter measures are beyond the scope of this paper. A straightforward way to measure IR performance for a P2P system is to compare the results to those that would be obtained from a centralized one. This is the approach taken by Neumann et al. [4] in producing a standardized benchmark framework for P2P systems, as well as Lue and Callan [6] to measure the performance of a hybrid P2P network. It is also the basis for PAC accuracy.

The above measures score documents equally, no matter where they appear in the result list. It is now widely acknowledged that taking into account the rank of documents in the result list can better model human perception. A number of rank-aware measures of IR performance have been proposed, including discounted cumulative gain (DCG) [2], and rank-biased precision (RBP) [3]. These two measures use the position model [7], which assumes a user will click on a search result if it is both relevant and if the user has examined it, where the probability of examination reduces the further down the list the result is. The cascade model is more sophisticated, and additionally takes into account the relevance of documents seen so far. Expected reciprocal rank (ERR) [8] is an example of a metric based on the cascade model.

The average ranked relative recall (ARRR) [9] and mean average overlap precision (MAOP) [10] measures are specifically designed to evaluate the effectiveness of P2P IR search, taking rank into account. They do so by comparing results to those that would have been obtained from a centralized system, and therefore are not reliant on human relevance judgements. The measure of rank-accuracy we present in this paper has a similar purpose to ARRR and MAOP, but is more flexible because it allows any weight to be assigned to a rank. It is also a natural extension to the PAC framework and is amenable to similar theoretical analysis.

We investigate how rank-accuracy can be increased by the non-uniform replication of documents across nodes. Cohen and Shenker [11] looked at different object replication policies to minimize the expected search length to find an object. They found that replicating objects across nodes in proportion to query rate did not, as might be expected, have an effect on average expected search length. Instead, replicating in proportion to the square root of query rate was found to be optimal. The PAC framework analyses a similar problem, but assumes a fixed search length. It was found that square root replication is not optimal, and a more complicated solution was derived using a convex optimization method [12]. Rank-aware replication is different to this earlier work on replication because in addition to document popularity, it also takes rank into account.

3 Rank-Accuracy

We assume an idealized network where all nodes operate correctly and there is no malicious behavior, as did the original work on PAC search. Node failures and security threats are important issues in P2P networks, but are beyond the scope of this paper. It is assumed there are n homogenous nodes in the network, there are m distinct documents in the collection and each node can store ρ documents. The total storage capacity of the network is R. Queries are sent to z randomly selected nodes, and relevant documents are combined and ranked to form a top-k result list. There are r_i copies of each document d_i replicated across nodes, such that $\sum_{i=1}^{m} r_i = R$. Multiple copies of the same document are not allowed on the same node.

The probability of finding c copies of a document d_i is binomially distributed and given by

$$P(c) = \binom{z}{c} \left(\frac{r_i}{n}\right)^c \left(1 - \frac{r_i}{n}\right)^{z-c} . \qquad (1)$$

It was shown in [1] that the probability $P(d_i)$ of finding at least one copy of document d_i is

$$P(d_i) = 1 - \left(1 - \frac{r_i}{n}\right)^z . \qquad (2)$$

In information retrieval typically there is not a simple one-to-one correspondence between query and documents. Instead, multiple documents of varying relevance can be retrieved and combined into a top-k list. We define *rank-accuracy* as a measure of correctness for the top-k result set for a query. Let $\mathcal{D}_k(j)$ be the set of top-k documents retrieved for query j from an exhaustive search of all nodes, and $\mathcal{D'}_k(j)$ be the set retrieved from a constrained search of z nodes. A weight $w_j(i)$ is assigned to each document d_i in $\mathcal{D}_k(j)$. The weight $w_j(i)$ is a function of the rank of document d_i in $\mathcal{D}_k(j)$, such that $\sum_{d_i \in \mathcal{D}_k(j)} w_j(i) = 1$. Various functions for $w_j(i)$ are possible and are discussed in Sect. 3.1. The rank-accuracy a_j for query j is then defined as

$$a_j = \sum_{d_i \in \mathcal{D'}_k(j)} w_j(i) . \qquad (3)$$

This can be compared to the rank-unaware accuracy measure for PAC search, which is defined [1] as

$$a_j = \frac{|\mathcal{D}_k(j) \cap \mathcal{D'}_k(j)|}{|\mathcal{D}_k(j)|} . \qquad (4)$$

If documents are assigned equal weights, so that $w_j(i) = \frac{1}{k}$, then it is easy to see that (3) and (4) are equivalent. The expected rank-accuracy $E(a_j)$ for a query j is given by

$$E(a_j) = \sum_{d_i \in \mathcal{D}_k(j)} P(d_i) w_j(i) = \sum_{d_i \in \mathcal{D}_k(j)} \left(1 - \left(1 - \frac{r_i}{n}\right)^z\right) w_j(i) . \qquad (5)$$

It follows that the average expected rank-accuracy A (when averaged over all queries), is

$$A = \sum_j q_j E(a_j) = \sum_j q_j \sum_{d_i \in \mathcal{D}_k(j)} \left(1 - \left(1 - \frac{r_i}{n}\right)^z\right) w_j(i) \qquad (6)$$

where q_j is the query rate of query j, such that $\sum_j q_j = 1$. For equal weighting, where $w_j(i) = \frac{1}{k}$, Equations (5) and (6) are equivalent to those derived in [12] for PAC search.

3.1 Rank Weightings

As discussed in Sect. 2, a number of techniques have been proposed to evaluate the quality of ranked top-k search results. Metrics such as nDCG and RBP assign greater weighting to documents appearing nearer the top of the result list, since these are assumed to be more important to the user. We propose using the same idea to assign weights for measuring rank-accuracy. In this paper we consider a weighting scheme derived from RBP [3]. Any scheme is possible, but RBP is sufficient to demonstrate the potential performance of rank-aware PAC search. For an RBP-like scheme, weighting W for a document at rank y is given by

$$W = (1 - p) \cdot p^{y-1} \tag{7}$$

where p models the persistence of the user and represents the probability that a user will go on to examine the next result in the list. This scheme can be used to assign values for $w_j(i)$. It has been shown that a value of p of around 0.6 or 0.7 is a reasonable approximation of user behavior [8]. RBP assigns greater weights to documents higher in the result list, and this skew increases as p decreases.

For a system where documents are replicated across nodes without regard to rank, average expected rank-accuracy A as given by (6) is unaffected by the weighting scheme. However, we shall see in the following section that A can be increased by using a replication policy that takes the ranked-weighting into account.

3.2 Rank-Aware Replication

For a given weighting scheme, we would like to choose a replication rate r_i to maximize A in (6). A simple, but sub-optimal policy is uniform replication. This involves distributing all documents onto the same number of nodes, so that r_i is given by

$$r_i = \frac{R}{m} \ . \tag{8}$$

For PAC search it was shown in [12] that average expected accuracy can be increased beyond that of uniform replication by replicating each document in proportion to its retrieval rate. A further improvement can be achieved by replicating in proportion to the square root of retrieval rate. Both policies increase average expected accuracy by increasing the replication of the more popular documents, although this is at the expense of the less popular documents. Here we propose similar techniques to boost rank-accuracy, but instead of basing replication on retrieval rate, we use weighted retrieval rate, where the weighting is determined by a document's rank in the top-k lists from exhaustive searches of all nodes. The intuition is that this will result in popular highly ranked documents being replicated more than popular documents that are ranked worse, and thus the average expected rank-accuracy, A will be improved. In practice, a representative query load may be unavailable to compute the required document distribution, but in a simulation in Sect. 5 we show that gains in rank-accuracy can still be achieved by replicating documents as queries are made.

We develop the rank-aware replication model by first defining an auxiliary set $\mathcal{V}$ that holds the weighted retrieval rate for each document, d_i, in the collection. Assuming the number of queries Q is finite, which is true in a limited period of time, then for each $v_i \in \mathcal{V}$, we have

$$v_i = \sum_{j=1}^{Q} q_j \cdot \zeta(j,i) \cdot w_j(i) \tag{9}$$

where

$$\zeta(j,i) = \begin{cases} 1 \text{ if document } i \text{ is in query } j\text{'s top-}k \text{ result list.} \\ 0 \text{ otherwise.} \end{cases} \tag{10}$$

We can replicate a document, d_i in proportion to its corresponding weighted retrieval rate, v_i or in proportion to the square root of v_i, in analogy with the replication policies proposed by Cohen and Shenker [11]. For rank-aware proportional replication, r_i is given by

$$r_i = R\frac{v_i}{\sum_i v_i} \tag{11}$$

and for rank-aware square root replication, r_i is given by

$$r_i = R\frac{\sqrt{v_i}}{\sum_i \sqrt{v_i}} \ . \tag{12}$$

It should be noted that these replication policies are restricted by the number of nodes n in the network. If (11) or (12) yields a value of r_i greater than n, then r_i is set to n and the unused capacity is allocated to the remaining documents. In Sect. 4 we shall see that (11) and (12) can achieve higher average expected rank-accuracy, A than uniform replication, but neither is optimal. We can find the optimum replication rate using a similar approach to [12]. To begin, (6) is expressed in closed form:

$$\begin{aligned} A &= \sum_j q_j \sum_i \left(1 - \left(1 - \frac{r_i}{n}\right)^z\right) \cdot \zeta(j,i) \cdot w_j(i) \\ &= \sum_i \left(1 - \left(1 - \frac{r_i}{n}\right)^z\right) \sum_j q_j \cdot \zeta(j,i) \cdot w_j(i) \\ &= \sum_i v_i \left(1 - \left(1 - \frac{r_i}{n}\right)^z\right) \ . \end{aligned} \tag{13}$$

Since r_i can only take integer values, finding the distribution of r_i to maximize A is an integer programming problem. An approximate solution to this problem was provided in [12] for PAC search using convex optimization, and we utilize the same solution here. The only difference is that v_i, here given by (9), includes the weighting term $w_j(i)$. Since the working is lengthy, we refer the reader to [12]. The solution yields

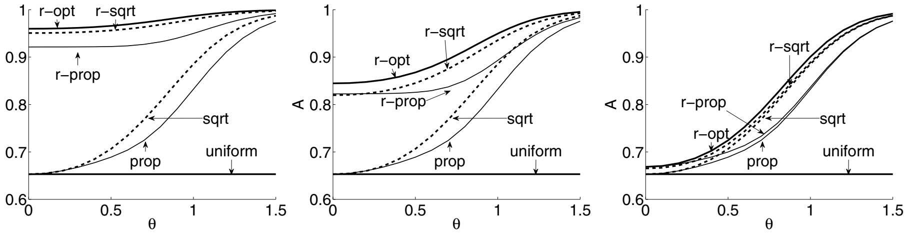
$$r_i = n - \frac{n(b'-1) - R'}{\sum_1^{b'-1} v_i^{-\frac{1}{z-1}}} \cdot v_i^{\frac{1}{z-1}} \tag{14}$$

where $R' = R - m + b' - 1$ and b' is an auxiliary variable chosen to enforce minimum and maximum values of r_i.

4 Theoretical Analysis

We now evaluate the theoretical effect of the rank-aware replication policies on rank-accuracy. In our analysis we assume there are $n = 10,000$ nodes. Each node stores $\rho = 500$ documents, and there are $m = 47,480$ distinct documents. Documents are replicated randomly across nodes according to the replication policy under test. It is assumed there are $4,748$ distinct queries that obey an inverse power law. The query rate q_j of query j is given by $\frac{1}{c}j^{-\theta}$, where c is a normalization constant so that $\sum_j q_j = 1$. We set $\theta = 0.7$ and the total volume of queries to 10,000. Studies have found that queries to web search engines typically follow such an inverse power law distribution, with exponent θ ranging between 0.7 and 1.5 [13]. Here we assume each query returns a top-10 result list that is disjoint i.e. each document only appears in the result list of one query, although each query may be repeated multiple times. Under this assumption it is easy to calculate the expected rank-accuracy for each query from (5), since the retrieval rate of each of the documents in the top-10 is simply the rate of the query. Using the RBP weighting scheme as given by (7), we compare the effect on rank-accuracy of different replication policies.

Figure 1 shows the effect of the query power law exponent θ on average expected rank-accuracy A, as given by (6), for three different RBP weightings, i.e. $p = 0.3, 0.6$ and 0.9. It is assumed that the query is sent to $z = 100$ nodes. There are a number of observation to be made. First, the more skewed the rank weighting, e.g. $p = 0.3$, the more pronounced is the gain in rank-accuracy for a rank-aware policy over the rank-unaware one. Conversely, when the rank weights are much less skewed, e.g. $p = 0.9$, i.e. there is a 90% probability that the user will look at the next document, the gain is much less. This is to be expected. Importantly, for rank weights set by $p = 0.6$, a value that was found to model typical user behavior, we observe significant improvements in rank-accuracy. Second, as the query distribution becomes more skewed, i.e. as θ becomes large, we observe that the performance difference across the various replication policies significantly decreases. This too is to be expected, as for very heavily skewed query distributions, rank-accuracy is dominated by just a few very popular queries and their corresponding result sets. Nevertheless, for θ values between 0.7 and about 1.25, and for $p = 0.6$, we observe significant differences across the various replication policies. Third, we note that even for a uniform query distribution, i.e. $\theta = 0$, all three sub-figures show a significant improvement in rank-accuracy when documents are replicated by one of the three rank-aware replication policies, i.e. *r-prop*, *r-sqrt* and *r-opt*. In contrast, replicating based on a rank-unaware policy for $\theta = 0$ produces no benefit over a simple uniform distribution of documents. That rank-accuracy improves for rank-aware replication policies even for a uniform query distribution is at first curious. However, this is due to our assumption that the top-10 query result lists are disjoint. Thus, 10% of documents have a rank of 1, 10% have a rank of 2, etc. Consequently, the 10% of documents with a rank of 1 are replicated more, than the 10% ranked at 2, and so on. As a result, the rank-accuracy is improved even for a uniform query distribution. Finally, we observe that in general a rank-aware replication policy based on the square root of weighted retrieval rate, *r-sqrt*, performs better than a rank-aware proportional policy, *r-prop*, and that the optimal rank-aware policy, *r-opt* always performs best.



(a) $p = 0.3$ (b) $p = 0.6$ (c) $p = 0.9$

Fig. 1. The effect of query power law exponent θ and different replication policies on average expected rank-accuracy A, using RBP weighting with (a) $p = 0.3$ (b) $p = 0.6$ (c) $p = 0.9$. In addition to uniform, there are curves for rank-unaware replication policies proportional (*prop*), square root (*sqrt*), as well as rank-aware policies for proportional (*r-prop*), square root (*r-sqrt*) and optimal (*r-opt*). For parameters $n = 10,000$, $\rho = 500$, $m = 47,480$, $z = 100$.

We also consider performance for individual queries. An inverse power law query distribution with exponent $\theta = 0.7$ is assumed. Figure 2 shows the expected rank-accuracy $E(a_j)$ for each query j, as given by (5) for $p = 0.6$. For the few most popular queries, we observe that a rank-unaware proportional replication policy, *prop*, actually provides the best expected rank-accuracy. However this gain comes at a significant expense - many more queries perform worse than a uniform distribution (horizontal line in figure). This indicates that documents retrieved by the most popular queries were replicated more than for other policies, but at the expense of less replication for other documents. The rank-unaware square root policy, *sqrt*, exhibits similar problems, though less pronounced than the rank-unaware proportional policy. In general, all three rank-aware policies perform better, with the square-root policy, *r-sqrt*, being superior to the proportional policy, *r-prop.* The optimum rank-aware policy, *r-opt*, exhibits the best performance, with all queries having better rank-accuracy than for a uniform replication policy.

5 Experimental Results

In order to confirm the preceding theoretical analysis, we performed three simulations on a network of $n = 10,000$ nodes. Each of the experiments progres-

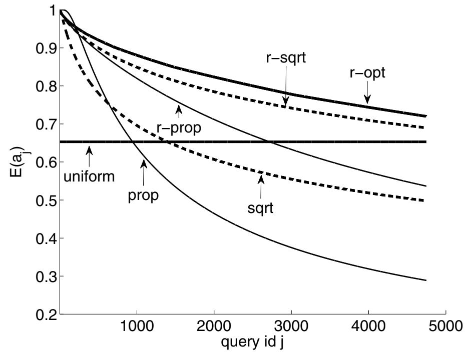


Fig. 2. Expected rank-accuracy $E(a_j)$ for each query j using RBP weighting with $p = 0.6$. In addition to uniform, there are curves for rank-unaware replication policies proportional (*prop*), square root (*sqrt*), as well as rank-aware policies for proportional (*r-prop*), square root (*r-sqrt*) and optimal (*r-opt*). For parameters $n = 10,000$, $\rho = 500$, $m = 47,480$, $z = 100$ and $\theta = 0.7$.

sively modeled a more realistic scenario. The results presented here are for rank-accuracy measured using an RBP weighting scheme with $p = 0.3, 0.6, 0.9$. Both rank-unaware and rank-aware proportional replication policies were tested, with the rank-aware policy using the same weighting scheme as used for measuring rank-accuracy. Due to lack of space we do not include results for the square root and optimal replication policies, although rank-accuracy for both these policies can be improved when rank-aware replication is used.

For Experiment 1, we assumed a rather artificial environment that exactly models the theoretical assumptions above. Specifically, we assumed (i) prior knowledge of the query distribution, (ii) the retrieval results (top-10) for each distinct query are disjoint, (iii) the global top-10 result list for each query is known and can be used by the replication policy, and (iv) the appropriate document replication, i.e. *prop* or *r-prop*, has been performed prior to performing the searches. We used a document collection of size $m = 47,480$. Each document was represented by an identifier, and these were distributed across nodes according to the replication policy under test. Each node stored $\rho = 500$ documents. For each artificial query we assigned a random top-10 set of document identifiers to act as the global result set that would be found from an exhaustive search of all nodes. These sets were disjoint. We issued $10,000$ queries, $4,748$ of which were distinct. The queries followed a power-law distribution with $\theta = 0.7$. Each query was issued to $z = 100$ randomly selected nodes. Each node returned the subset of document identifiers present in its index that matched the corresponding identifiers in the query's associated global result set. These subsets were combined at the query node to form a single ranked result list for the query. This list was used to compute the rank-accuracy for the search. These results were then averaged across all the 10,000 issued queries to produce a single average rank-accuracy score. The average rank-accuracies produced by the simulation for rank-unaware / rank-aware policies were 0.71/0.93, 0.71/0.83, 0.71/0.72 for $p = 0.3, 0.6, 0.9$ respectively, which correspond to increases of 31.0%, 16.9%, 1.4%. These results show that for all values of p, average rank-accuracy is increased when rank-aware replication is used. The improvements are greater the lower the value of p. The improvements closely match the theoretical results predicted in Fig. 1 for $\theta = 0.7$.

For Experiment 2, queries were drawn from a Yahoo! web search log [14]. For each query there is an anonymized query identifier along with anonymized document identifiers corresponding to each of the ranked top-10 documents displayed to the user. The first $1,000,000$ queries were used for the simulations. From the results of each query, $m = 461,788$ distinct document identifiers were extracted and used to represent the document collection. Each node had a storage capacity of $\rho = 1,500$ document identifiers, giving a total network storage capacity of $R = 15,000,000$. In this simulation, the result sets were no longer disjoint. However, we still assumed (i) the global top-10 result list for each query is known and can be used by the replication policy, and (ii) the appropriate document replication has been performed prior to performing the searches. The average rank-accuracies produced by the simulation for rank-unaware / rank-aware policies were 0.72/0.86, 0.69/0.76, 0.64/0.65 for $p = 0.3, 0.6, 0.9$ respectively, which correspond to increases of 19.4%, 10.1% and 1.6%. As with Experiment 1, rank-aware replication increased average rank-accuracy, with the improvement greater for lower values of p. However, the improvements were not as large as with Experiment 1. This can be attributed to the more skewed Yahoo! query distribution. The Yahoo! queries exhibited an approximate power law distribution with $\theta \approx 0.7$, but there were more extremely popular queries than for the first simulation. Interestingly, as p decreased, average rank-accuracy increased for rank-unaware replication. This is not predicted by the theoretical model or found in the previous simulation, and is due to a small correlation between rank and the number of queries a document is relevant to.

For Experiment 3, we used the same parameters as Experiment 2, but no longer assumed (i) the global top-10 result list for each query is known and can be used by the replication policy, and (ii) the appropriate document replication has been performed prior to performing the searches. Instead, documents were initially uniformly randomly distributed across nodes. On retrieving the top-10 documents for a query (which may only be a subset of the global top-10 documents that would have been found from searching all n nodes), the querying node replicated the documents onto up to 20 other nodes in accordance with the rank-unaware / rank-aware policy. As queries were performed, the distribution of documents moved away from uniform towards rank-unaware / rank-aware proportional. The values of average rank-accuracy produced by the simulation for rank-unaware / rank-aware policies were 0.62/0.78, 0.58/0.69, 0.54/0.56 for $p = 0.3, 0.6, 0.9$ respectively, which correspond to increases of 25.8%, 19.0% and 3.7%. Overall, average rank-accuracy for all policies was lower than for Experiment 2. This is expected because rank-unaware / rank-aware proportional replication is only approximated.

6 Conclusions and Future Work

Evaluation of IR performance in unstructured P2P architectures often considers the proportion of documents retrieved in comparison to an exhaustive search. The PAC framework uses such a measure to model probabilistic search. However, it does not consider the rank order of the documents in the result set, despite the fact that the rank order is known to significantly affect user perception of IR performance. To address this, we proposed a rank-weighted measure of accuracy. The weighting can follow one of the many rank-based evaluation metrics, e.g. DCG, RBP.

Previous work has shown that the expected search length and the rank-unaware accuracy can be significantly improved by replicating documents non-uniformly based on the query distribution. Building on this work, we proposed a rank-aware replication policy to increase rank-accuracy, replicating documents across nodes based on retrieval rate, but weighted by their corresponding rank in queries.

We analyzed the performance of an RBP-like scheme that assigned a greater weighting to documents appearing nearer the top of the result list. Theoretical modeling showed that rank-aware replication can achieve higher rank-accuracy when averaged over all queries than the rank-unaware replication of PAC. This improvement was greater the more skewed the weighting scheme and the less skewed the query distribution. An idealized simulation confirmed our theoretical analysis. We also ran simulations using real queries drawn from the Yahoo! web search engine. When documents were distributed based on prior knowledge of the query distribution, average rank-accuracy was increased by 19.4%, 10.1% and 1.6% for RBP with $p = 0.3, 0.6, 0.9$, when compared to rank-unaware replication. When no prior knowledge of the query distribution was assumed, and documents were distributed as queries were made, both rank-unaware and rank-aware replication achieved lower absolute values of rank-accuracy. However, rank-aware outperformed rank-unaware replication by 25.8%, 19.0% and 3.7% for $p = 0.3, 0.6, 0.9$. We would expect greater improvement for a less skewed query distribution. In practice, for a large-scale system with a large document collection and a huge number of queries, it may be infeasible to compute the required document distribution based on prior queries. Therefore, in future work we intend to build upon the technique used in Experiment 3 and investigate further how a rank-aware distribution of documents can be achieved by replicating documents as queries are made.

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

References

1. Cox, I.J., Fu, R., Hansen, L.K.: Probably Approximately Correct Search. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 2–16. Springer, Heidelberg (2009)

2. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (2002)
3. Moffat, A., Zobel, J.: Rank-biased precision for measurement of retrieval effectiveness. ACM Transactions on Information Systems (TOIS) 27(1), 2 (2008)
4. Neumann, T., Bender, M., Michel, S., Weikum, G.: A reproducible benchmark for P2P retrieval. In: Proc. ACM Wkshp. Exp. DB (2006)
5. Yang, Y., Dunlap, R., Rexroad, M., Cooper, B.: Performance of full text search in structured and unstructured peer-to-peer systems. In: IEEE INFOCOM, pp. 2658–2669 (2006)
6. Lu, J., Callan, J.: Content-based retrieval in hybrid peer-to-peer networks. In: CIKM 2003: Proceedings of the 12th International conference on Information and Knowledge Management, New Orleans, LA, USA (2003)
7. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 87–94. ACM (2008)
8. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 621–630. ACM (2009)
9. Witschel, H., Holz, F., Heinrich, G., Teresniak, S.: An Evaluation Measure for Distributed Information Retrieval Systems. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 607–611. Springer, Heidelberg (2008)
10. Lu, J., Callan, J.: User modeling for full-text federated search in peer-to-peer networks. In: SIGIR 2006: Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA (2006)
11. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 177–190. ACM (2002)
12. Fu, R.: The quality of probabilistic search in unstructured distributed information retrieval systems. PhD thesis, University College London (2012)
13. Baeza-Yates, R., Gionis, A., Junqueira, F., Murdock, V., Silvestri, F.: The impact of caching on search engines. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, ACM (2007)
14. Yahoo!: Yahoo! webscope dataset anonymized Yahoo! search logs with relevance judgments version 1.0, http://labs.yahoo.com/Academic_Relations