Ranking Information Resources in Peer-to-Peer Text Retrieval: an Experimental Study

Hans Friedrich Witschel
NLP department
University of Leipzig
witschel@informatik.uni-leipzig.de

ABSTRACT

This paper experimentally studies approaches to the problem of ranking information resources w.r.t. user queries in peer-to-peer information retrieval. In distributed environments, for each given user query and a set of information resources that are available, we need to select the right subset of these resources to forward the query to. Here, we study the problem of pruning descriptions of resources to acceptable lengths in a peer-to-peer scenario and two approaches to overcome the mismatch problem that may arise as a consequence of the pruning, namely query expansion and learning better resource descriptions from query streams. The results show that resource descriptions can be pruned to a large extent without ill effects and that learning better descriptions from query streams works much better than query expansion.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Selection Process, C.2.4 [Distributed Systems]: Distributed Applications.

General Terms: Algorithms, Experimentation.

Keywords: Peer-to-peer information retrieval, profiles, query expansion, intelligent query routing, profile adaptation.

1. INTRODUCTION

In distributed information retrieval (DIR, [5]), a central instance, often called *broker* receives user queries, forwards them to a selection of IR databases and then merges the results returned by these into a final ranking.

In peer-to-peer information retrieval (P2PIR), there is no central instance (cf. e.g. [11]). Rather, each peer has knowledge of the addresses of a small number of other peers, its neighbours. In order to retrieve information in a P2PIR system, a peer that receives a query searches its local database for matching data items and then forwards it to a subset of its neighbours, which will proceed in the same way until some stopping criterion is met. The entire process of forwarding user queries in a P2P network is often called *query routing*.

The task of selecting only a subset of information resources from all available ones is motivated – in both cases – by the wish to reduce costs: in DIR, selecting a large number of resources is costly because it may take a long time and cause overload on the selected databases – which may actually charge the user for each returned result.

In P2PIR, cost is most often measured in the number of messages that is generated by a query. A high number of messages results in the underlying physical network becoming slow and overloaded.

1.1 Profiles

In order to perform the selection task, each entity – be it a broker in DIR or a peer in P2PIR – needs to have a description of the content offered by each of the available information resources. Queries will be matched against these *profiles* in order to make a selection decision.

In DIR, it is common to represent information resources by so-called *unigram language models*, that is, the set of terms (or a subset of these) that occur in the documents of the resource, together with their document frequencies or some similar statistics [7, 12]. The idea of most of these approaches is to treat information resources as giant documents and to use a retrieval function, identical or similar to the ones used for documents, to rank these resources and select the top-ranked ones.

In P2PIR, there is no such commonly accepted peer representation. Although the representation of peers by unigram language models is also used [19, 9], a number of alternatives exist, including the use of categories from *ontologies* or *taxonomies* to represent peers (e.g. [4]) or approaches that have each peer record which other peers have (successfully) answered which queries, resulting in "distributed" profiles (see section 2.2 below).

The main reason for the emergence of these alternatives is the need for *compactness* in P2PIR: peer profiles often need to be sent around to other peers and stored in their routing tables. Bandwidth and storage limitations present in P2P settings make it necessary for profiles to be very compact.

1.2 Problem definition

We will concentrate on unigram language models as peer profiles in the experiments of this work. Starting from this setting, the need for compactness of peer profiles implies that it may not be possible to store all the index terms occurring in a peer's document collection in its profile. Instead, we need to select a subset of all those terms, in a way that still

LSDS-IR'08, October 30, 2008, Napa Valley, California, USA.
Copyright 2008 ACM 978-1-60558-254-2/08/10 ...$5.00.

allows the routing algorithm to predict which peers are most likely to offer the desired content w.r.t. a given query. But even using the most elaborate selection of profile terms, we will inevitably lose information as profiles get smaller.

It is the aim of this work to explore this trade-off:

- Starting from simple profiling and matching techniques, the first question is: how does the degradation in retrieval effectiveness correlate with profile compression? That is, how many terms can we prune from a profile and still have acceptable results?
- In a second stage, the initial profiling and matching strategies will be refined: techniques for both *learning better profiles* from query streams and *refining queries* by query expansion will be compared against each other.

When applying techniques such as query expansion, new challenges arise in a distributed setting: as there is no global view on the data, we cannot access all the documents of a distributed collection for performing e.g. pseudo feedback.

The rest of this paper is organised as follows: section 2 presents related work that has been done in the area of resource description and selection both in DIR and P2PIR. Section 3 defines the techniques that will be compared to each other. The experimental setting used to perform this comparison is described in section 4; section 5 presents the results of this comparison before section 6 summarises them.

2. RELATED WORK

2.1 Pruned profiles

The first of the two problems introduced above – namely the question of how strongly we can compress peer profiles and still have acceptable recall – has received relatively little attention in both the DIR and the P2PIR community. In P2PIR, this may be because many approaches do not use unigram language models for representing peers. One exception is [3] where various pruning algorithms are presented. However, the work in [3] does not consider absolute thresholds for profile sizes as we do here, but varies parameters of pruning algorithms, leading to varying, but unpredictible profile sizes.

In DIR, on the other hand, resource descriptions are usually stored on large broker servers where space constraints are not a major problem. However, some work has been done on that topic [18], pruning index terms from long documents. However, the trade-off between profile size and retrieval effectiveness is not evaluated systematically.

2.2 Resource description: profile refinement

The basic idea of profile refinement is to characterise a peer or information resource not only by the content that it offers, but by the queries for which it provides relevant documents.

Work on query-adapted profiles is rare in DIR, some related ideas can be found in early work on *collection fusion* [25] and adaptive resource selection [13].

In P2PIR, on the other hand, many systems use what could be called a *collective* discovery approach by having every peer in the system store query-related information associated with other peers. The entirety of routing table entries pointing to a peer can be viewed as its profile, shared throughout the community.

Collective discovery approaches either store keywords from queries [14, 1] or full queries [15, 16] in routing tables, together with the addresses of peers that provided the answer. Social metaphors are the basis for building profiles in [17]. An explicit *learning* approach is introduced in [1] and a semi-supervised learning approach is described in [21].

2.3 Resource selection: query refinement

Another way to improve resource selection is to refine the queries instead of the profiles, e.g. by query expansion. As is pointed out in [29], query expansion may help to overcome problems due to the loss of document boundaries in profiles.

Two studies [29, 20] examine the effectiveness of query expansion in DIR, reaching rather different conclusions: the first study finds significant improvement over the baseline CORI selection, the second one has discouraging results.

Query expansion is also used in some approaches to P2PIR: in [8], a local pseudo feedback approach based on language modeling is presented, first ranking peers w.r.t. the unexpanded query and then using the best k results returned by the top-ranked peer for pseudo feedback.

A final optimisation of resource selection concerns not so much the query formulation process, but the overlap of documents among information resources. This problem is addressed in DIR by [24] and in P2PIR by [2], but not in the experiments of section 5. However, it will be interesting to investigate overlap more closely in future work.

2.4 Contribution

The contribution made by this work is twofold:

- So far, the trade-off between absolute profile size and retrieval effectiveness has not been evaluated thoroughly for P2PIR. The studies in DIR [18, 6] explore just one or two values for profile size and the only study in P2PIR [3] does not use absolute profile sizes.
- So far, many advanced solutions to collection selection and query routing have been proposed and most of them have been evaluated in isolation. There have been comparative evaluations in DIR [10], but, to the best of my knowledge, this is the first evaluation that compares a selection of approaches against each other in a unified P2PIR evaluation setting, including methods for profile adaptation that have not been explored in DIR.

3. SOLUTIONS TO BE EXPLORED

This section presents the approaches for solving the peer selection problem in P2PIR that will be explored in the experimental section below.

3.1 Preliminaries

Before we can start with the actual profile and query refinement strategies, some preparatory issues need to be fixed:

- *Profiles*: The computation of profiles is designed to allow for a variant of the CORI algorithm for ranking information resources (cf. [5]): each term t that appears in a peer p's document collection is assigned a weight according to the CORI formula

$$P(t|p) = d_b + (1 - d_b) \cdot T \cdot I \qquad (1)$$

where the minimum belief component d_b is set to 0. The I component is normally computed using the number of resources (i.e. peers in our context) that contain term t. Since this is unknown in a P2PIR scenario, it will be replaced by an idf weight as discussed below. The T component is computed (as in CORI):

$$T = \frac{df_t}{df_t + K} \tag{2}$$

$$K = k \cdot ((1-b) + b \cdot \frac{cw}{avgcw} \tag{3}$$

where df_t is the document frequency of term t within p's collection, cw is the number of index terms in p's collection and $avgcw$ is the average number of terms in all peer collections[1] and k and b are free parameters, set to 100 and 0.75 in the experiments, respectively.

- *Compression*: Profiles are compressed by simple thresholding applied to the list of terms ranked by CORI weights, i.e. the n terms ranked most highly by $P(t|p)$ will form the profile of peer p. In the experiments, the n-values 10, 20, 40, 80, 160, 320 and 640 are explored and compared to using uncompressed profiles. The sizes of profiles are absolute, because we must assume that the maximum acceptable size of a profile is defined by some technical constraints dictated by the underlying network.

- *Global term weights*: Since no global view on the collection is permitted, the I component of the CORI algorithm cannot be computed. It will be replaced by idf estimates derived from a mix of a large external collection and a small sample of the target collection as described in [26]. The same idf estimation is used when performing the centralised retrieval, so that scores of documents are the same in both scenarios. This means that differences in scores among the two will only be attributable to the query routing decisions – and not blurred by any result merging effects. In addition, as pointed out in [26], such idf estimation can realistically be implemented in P2PIR systems.

- *Query routing and retrieval*: For query routing, peers will be ranked by the sum of the CORI weights of all the query terms that are contained in the (pruned) profile. Each peer that receives the query retrieves documents from its local repository using the BM25 retrieval function [22].

3.2 Baselines

Next, we define some baselines that the advanced query routing strategies can be compared to:

- *Random*: Rank peers in random order.

- *By size*: instead of creating a content-related profile for each peer, just rank peers by the number of documents they hold.

- *Base CORI*: apply the procedure described in the previous section, refining neither queries nor profiles.

3.3 Query expansion

All query expansion methods used in experiments are based on local context analysis (LCA, see [30]) in a slightly modified version. There are two changes w.r.t. the original LCA formulation in [30]

- The computation of idf values is done as described above.

- In the original work on LCA, expansion concepts are defined as noun phrases. We relax this definition and allow arbitrary index terms as concepts.

The general expansion strategy LCA is used with three types of expansion collections:

- *The web*: queries are passed to the API of the Yahoo! web search engine and the top 10 results are retrieved. The "passages", from which expansion terms are taken, are the snippets that the search engine returns as a summary of the result page.

- *Local pseudo feedback*: As an alternative to these global resources, a local expansion strategy as described in [8] is implemented that first ranks peers using the original query, retrieves the 10 best results returned by the top-ranked peer and feeds them into LCA.

- *Global pseudo feedback*: Instead of using documents only from the top-ranked peer, this strategy assumes knowledge of the whole distributed collection and uses the 10 best results that a centralised system would return. This strategy serves as an upper baseline for the other two query expansion strategies: although it cannot be applied in a P2P setting, it can serve to show how effective pseudo feedback could be if we had complete knowledge.

3.4 Profile adaptation

Adapting profiles is done using a simple learning rule inspired by the reinforcement learning in [28, 1].

The idea behind that approach is to boost the weight $w_{i,p}$ of a query term i in a peer p's profile if p has high-quality results for the query (note that formula 4 only changes weights for terms that are already in p's profile. In the future, it may be interesting to study a variant where new terms may be added to profiles). The learning rule used in this work is as follows:

$$w_{i,p}(t+1) = \Big(\frac{\text{RP@}k(D_p, D_o) + 1}{\text{AVGRP} + 1}\Big) w_{i,p}(t) \tag{4}$$

where RP@k stands for "relative precision" at k documents, a measure for the quality of a ranking as defined below. D_p is the result list returned by peer p, D_o is the result list returned by all other peers the query has reached. AVGRP is the average over all RP values of those peers. In the experiments below, $k = 10$ was used throughout.

For now, it is sufficient to know that RP measures how highly (on average) the results in D_p are ranked in D_o. Hence, it is a measure of the quality of the results returned by peer p that is solely based on the *ranks* of those result documents in a reference ranking D_o.

As an example, consider the query "white house" and a peer p returning a ranking $D_p = [d_1, d_2]$ of two documents. Now, p learns of the results D_o of all other peers that have

[1] This is again unknown in a P2PIR setting, but was taken from the whole collection for the experiments. The effect of using only a rough estimate for this quantity remains to be examined in future work.

contributed to the query; based on this knowledge, p computes $RP@k(D_p, D_o)$ as a measure of quality of its own results, as well as the average RP value $AVGRP$ taken over all contributing peers' results. Now, if $RP@k(D_p, D_o)$ is greater than $AVGRP$, p will increase the weight of the terms "white" and "house" in its profile as prescribed by equation 4.

In practice, the learning is performed on a query log being partitioned into a training and a test set. During training, we assume – optimistically and merely for the purpose of evaluation – that each training query reaches *all* peers and that hence D_o consists of all documents found by a centralised system.

For each peer p that possesses at least one document $d \in D_o$, we compute the new weight of query terms in p's profile as given in equation 4. The update of $w_{i,p}$, however, is only executed if the ratio $\frac{\mathrm{RP@}k(D_p,D_o)+1}{\mathrm{AVGRP}+1}$ is greater than 1, that is when p's results are better than the average.

Note that in a real P2P system, when peers manage their own profiles, this procedure requires either that the query is sent back along the path it was initially routed, with the result set D_o attached to it, in order for each peer to be able to compare its own results to that of the others. Alternatively, the querying peer – having received the results – may compute scores for peers and notify those with a ratio of $\frac{RP@k(D_p,D_o)+1}{AVGRP+1}$ greater than 1.

For that purpose, D_o should be pruned to a reasonable size. It is sufficient for D_o to consist of document hashes – so that peers can identify the rank of their own documents within D_o.

Since the weights $w_{i,p}$ may grow exponentially large with this approach, the final weights $w'_{i,p}$ in peers' profiles are obtained by rescaling with a logarithm: $w'_{i,p} = \log(1 + w_{i,p})$. This way of rescaling was found to work best in a preliminary set of experiments. Its main advantage is the fact that $w'_{i,p} \approx w_{i,p}$ for $w_{i,p} \ll 1$. Since this applies to most unadapted weights, rescaling has very little effect on these, while smoothing some of the adapted weights that have grown very large.

After training is completed, query routing is performed by matching queries from the test set against the adapted profiles. The test set is identical to the queries used to evaluate all other strategies (that is, the baselines and the query expansion methods).

4. EXPERIMENTATION ENVIRONMENT

4.1 Simplifications

Here, I will introduce a few choices of parameters that were fixed in the experiments.

First, we assume that each peer truthfully creates and manages its own profile, i.e. profiles are accessible at one point (namely their source) and not shared throughout the community.

This assumption allows for the most important simplification that is being made in this work: in an attempt to study the query routing problem in isolation – independent of overlay topology – we only evaluate a DIR scenario, no real P2PIR simulation is performed. Equivalently, we could say that we assume the overlay network to be a complete graph, i.e. each peer has complete knowledge of all other peers' addresses and profiles (as in [9]).

Apart from the wish to decouple neighbour selection and query routing, this decision is expected to help reduce the number of free parameters considerably: when trying to simulate a P2P community, we need to make assumptions regarding not only the topology of the overlay network, but also the distribution of queries among peers, whether or not forwarding to more than one peer is allowed, churn (i.e. whether or not a contacted peer is on-line or not) etc.

However, the claim is made that the results obtained in the experiments below are valid not only for DIR, but also (and even more so) for P2PIR. In fact, by not committing to particular settings of P2PIR parameters, we can expect the results obtained to be valid across a large number of P2PIR systems with very different settings of these parameters.

On the one hand, the above claim is based on the assumption that a query routing algorithm that performs well in a situation where all peers' profiles are known – i.e. in DIR – will also do so when applied to only a subset of these – as is typically the case in P2PIR. On the other hand, care is taken to design characteristics of the DIR simulation in a way that is typical for P2PIR scenarios – as *opposed* to DIR scenarios:

- Profiles are pruned (with varying sizes). This is untypical in DIR because there are normally no size restrictions for resource descriptions.
- In DIR, evaluations usually use at most a few hundred information resources, in P2PIR we want to use far more peers, at least a few thousand.
- While information resources in DIR are normally large and semantically heterogenous, peers can be expected to share a smaller amount of documents belonging only to a few selected topics (where the exception of a few very large and heterogeneous peers proves the rule).

4.2 Test collections

We will examine two application scenarios, represented by the CiteSeer collection on the one hand and the Ohsumed and GIRT collections on the other. The latter two were chosen because their documents are labeled with categories that may easily be identified with peers.

4.2.1 CiteSeer

The first scenario is one in which individuals within a certain community share their own publications. We will use the CiteSeer database. Authors are identified with peers, i.e. a peer shares the documents that the corresponding person has (co-)authored. A query log is used for extracting test queries.

The collection was downloaded from the CiteSeer web site[2] on 17th November 2005 and contains – after removing duplicates – 441,178 abstracts from which a total of 230,922 distinct author names was extracted and identified with peers.

The distribution of the number of papers per author follows a power-law, which means that the majority of peers (149,421 = 64.7%) has only one or two documents. This distribution is consistent with measurements of peer size distributions in real P2P file-sharing networks [23].

The queries used in the experiments are taken from a portion of the CiteSeer's access logs, dating from August and

[2] http://citeseer.ist.psu.edu/oai.html

September 2005. This portion contains 712,892 successive queries, after deleting queries that were obviously generated by bots. Among these queries, there are 367,110 distinct ones, of which 122,082 occur more than once. Frequency of queries follows another power-law, i.e. there are a few very frequent queries and a vast majority of rare ones.

For the experiments, the last 10,000 queries of the log were used to evaluate all strategies. The first 702,892 queries were used as a training set in the evaluation of profile adaptation.

4.2.2 *Ohsumed and GIRT*

The second scenario is a situation where digital libraries specialised in various subdisciplines are joined together. In order to evaluate this, two test collections were used, Ohsumed and the German GIRT-4 collection. In both cases, documents have classes from a thesaurus-like resource assigned to them which are identified with peers, i.e. a peer shares all documents that are classified under the corresponding category. In case of Ohsumed, the classes are so-called MeSH terms (Medical Subject Headings), in case of GIRT controlled terms from a thesaurus for the social sciences.

In the Ohsumed collection, documents have an average of 10.6 MeSH terms assigned to them (standard deviation is 4.3), in GIRT documents have an average of 10.15 controlled terms (standard deviation 3.99). The distribution of peer sizes – when identifying MeSH terms or controlled terms with peers – again resembles a power law in both cases. In Ohsumed, there is a total of 14,623 MeSH terms in the collection, of which only 7,124 have more than 40 documents. In GIRT, we have 7,151 distinct controlled terms, with 3,847 having more than 40 documents.

This means that, in both cases, a significant number of peers will share only few documents. This might seem unrealistic for a digital library scenario. On second thought, however, there are also reasons for assuming the contrary: if we allowed digital libraries to be linked in a peer-to-peer fashion, it is not unlikely that their sizes would evolve into a power-law distribution, a phenomenon which is almost ubiquitous in real-life distributions.

4.3 Evaluation measures

For the Ohsumed and GIRT collections, existing relevance judgments are used so that the possibility of the distributed system performing better than a centralised one is not ruled out.

For the CiteSeer collection, we do not have relevance judgments. Therefore, the performance of the distributed retrieval system will be evaluated by comparing its results to that of a centralised system via the measure relative precision (RP). It exploits the ranking of the centralised system as an indicator of probability of relevance and is defined as follows:

Let C be the ranking of a reference (e.g. a centralised) retrieval system for a given query. Then we estimate the probability of relevance of a document d as the inverse of its rank $r_C(d)$ in C:

$$p(rel|d, C) = \frac{1}{r_C(d)} \qquad (5)$$

With this notion of probability of relevance, we can define the new measure relative precision at k documents for a ranking D returned by a distributed system as the average probability of relevance $p(rel|d, C)$ among the first k documents in D:

$$\mathrm{RP@}k(D, C) = \frac{1}{k}\sum_{i=1}^{k} p(rel|d, C) = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{r_C(d)} \qquad (6)$$

Later, experiments will be performed with $k = 10$, mostly because it is common (e.g. in web search) for search engines to display the first 10 results on the first page and because very few users look at the remaining result pages.

For a motivation of RP, see [27] where a predecessor of RP with very similar properties is introduced.

5. EXPERIMENTAL RESULTS

5.1 Basic evaluation procedure

The basic procedure applied in all evaluations of this section is to judge the quality of a peer ranking by the quality of the results that will be retrieved if peers are visited in the order implied by the ranking.

The top 100 peers are visited according to the peer ranking, and effectiveness of the resulting merged *document* ranking (the best 1,000 documents found so far) is measured after visiting each peer. In all cases, if there is no peer left with a score greater than 0, the next best peer is chosen randomly.

Because document scores are comparable across all peers (cf. section 3.1), merging rankings is trivial: when visiting peer i, its set of documents is united with the documents found at peers $1, .., i-1$ and the resulting set of documents is sorted by the documents' global scores and pruned to a length of 1,000.

5.2 Profile pruning

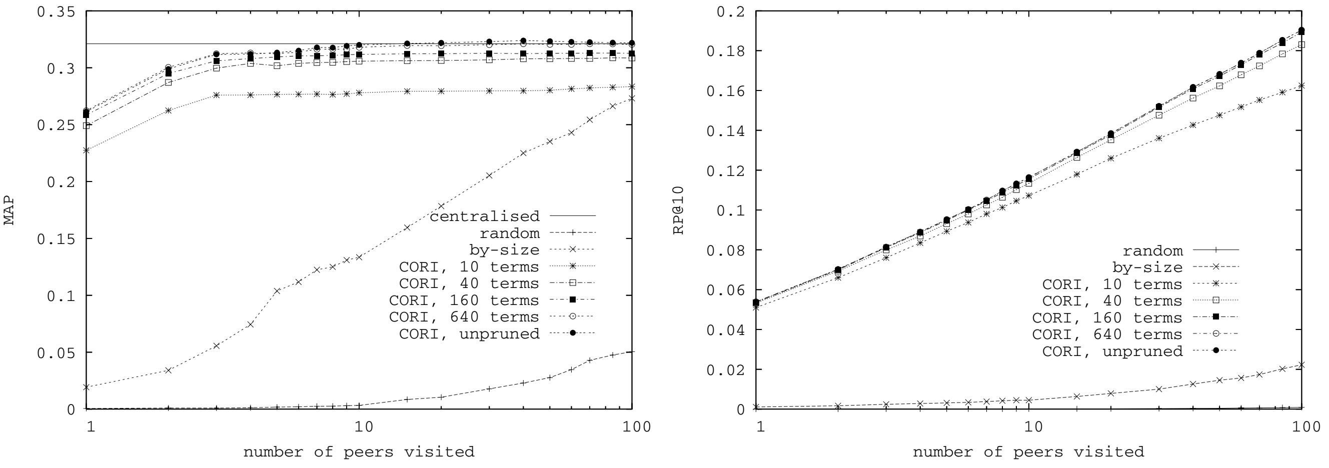
We now turn to the first of the two questions formulated in section 1.2: how strongly does degradation of retrieval effectiveness correlate with profile compression? To answer this question, we will compare the effectiveness of runs with pruned profiles to that reached with unpruned profiles.

This comparison will be based on statistical significance testing (using a Wilcoxon test on a 95% confidence level) in the case of GIRT and Ohsumed. For Citeseer, I chose to abandon significance tests because tests with such a vast number of queries (10,000) will always yield significant differences. Instead, a difference between a pruned-profile run and the unpruned baseline is considered meaningful iff the value of the corresponding measure is *not* within 5% of that of the baseline.

Figure 1 shows the performance of pruned-profile runs for some profile sizes as a function of number of peers visited. Table 1 shows the degradation of effectiveness introduced by profile pruning when compared with unpruned profiles (this is done only for the few top peers (5 and 15) because these are the most interesting ranges in P2PIR).

We see that, for Ohsumed and Citeseer, the majority of measurements yield no meaningful differences between pruned and unpruned profiles from 20 terms on.

For GIRT, the situation is a little more complicated: pruned profiles seem to deliver better results on average within the first 5 peers than within the first 15 peers, i.e. pruned-profile results seem to get worse – in comparison with the unpruned run – as more peers are visited. In all cases, profiles need to consist of at least 80 terms in order to reach the effectiveness of unpruned profiles. When comparing the effectiveness of the pruned-profile run to that of a centralised system, how-



(a)

(b)

Figure 1: Effectiveness as a function of number of peers visited for baseline runs with various profile sizes for (a) Ohsumed in terms of MAP and (b) CiteSeer in terms of RP@10. For Ohsumed, effectiveness of the centralised system is given for comparison. RP@10 is used for CiteSeer instead of MAP because no relevance judgemnts are available there.

	Ohsumed		GIRT		Citeseer	
Profile size	15	5	15	5	15	5
10 terms	-1	-1	-1	-1	-1	-1
20 terms	0	0	-1	-1	0	0
40 terms	0	0	-1	-1	0	0
80 terms	0	0	-1	0	0	0
160 terms	0	0	-1	0	0	0
320 terms	0	0	-1	0	0	0
640 terms	0	0	0	0	0	0

Table 1: Pruned profiles: the entries are -1 if, among the first 15 and 5 measurements, respectively (a measurement taking place after each visited peer), there is a majority of measurements stating that the pruned profiles yield results that are significantly worse than when using unpruned ones; the entry is 0 if the majority of measurements yield no significant difference between pruned and unpruned profiles.

ever, we find that – when a profile consists of 80 or more terms – results are not significantly worse than those of the centralised system after visiting a maximum of 4 peers. This implies that, although there may be significant differences w.r.t. the unpruned-profile runs, these differences do not exist w.r.t. the centralised run and that hence the results delivered using pruned profiles will definitely be acceptable from 80 terms on.

To see the impact of pruning in terms of how much compression it yields, we study the space savings achieved by the various profile sizes given in table 2. The average Ohsumed and GIRT peers have large profiles so that profile pruning has large impact. For Citeseer, where most peers have smaller profiles anyway, the impact is much smaller.

These results are interesting because they suggest that the degradation in effectiveness that is caused by pruning profiles to a predefined absolute size does not necessarily depend on the original size of profiles: although most unpruned Ohsumed profiles are large, they can be pruned heavily without ill effects. GIRT profiles – although of similar size – are harder to prune, for reasons probably related to term-distributional phenomena.

Profile size	Ohsumed	GIRT	Citeseer
10 terms	99.97%	99.76%	93.90%
20 terms	99.93%	99.52%	87.84%
40 terms	99.86%	99.05%	76.08%
80 terms	99.73%	98.10%	56.62%
160 terms	99.46%	96.25%	37.35%
320 terms	98.94%	92.69%	19.50%
640 terms	97.97%	86.17%	6.80%

Table 2: Space savings for profile pruning: the values are obtained by dividing the total number of terms that are used in pruned profiles of the various sizes by the number of terms that occur in unpruned profiles and subtracting that figure from 1.

All in all, we conclude that pruning profiles does not degrade results nearly as much as one might expect.

5.3 Query expansion

In this section, we will study the question whether query expansion can improve peer rankings. To that end, we expand queries using the three strategies discussed in section 3.3 and use the expanded queries with the CORI selection algorithm to obtain a peer ranking. Expansion is only used for ranking peers, *not* for retrieving and ranking documents. This means that the scores of documents remain the same as in the experiments above.

Table 3 shows the effectiveness of expanded queries as compared to that of unexpanded ones for Ohsumed. This is done by analysing performance within the first 15 peers visited and giving all intervals M where expanded queries perform significantly better than unexpanded ones and all intervals M' where they perform significantly worse.

	Ohsumed						**Citeseer**					
	Web expansion		*Local Feedback*		*Global Feedback*		*Web*		*Local*		*Global*	
Profile size	M	M'	M	M'	M	M'	M	M'	M	M'	M	M'
10 terms	[5,5]	–	–	[6,6], [12,13]	[5,15]	–	–	[1,15]	–	[1,15]	[1,15]	–
20 terms	[4,7]	–	[10,11],[15,15]	[1,1]	[3,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
40 terms	–	[1,1]	[9,15]	[1,1]	[4,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
80 terms	[3,3]	[1,1]	[13,15]	[1,1], [11,12]	[4,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
160 terms	[3,3], [5,6]	[1,1]	[3,3]	[1,1], [12,15]	[3,6],[8,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
320 terms	–	[1,1]	[4,4]	[11,11],[13,15]	[5,5],[10,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
640 terms	–	[2,2]	[4,5],[7,8],[10,15]	–	[8,8],[11,15]	[1,1]	–	[1,15]	–	[1,15]	[1,15]	–
all terms	–	[1,1]	[8,9]	[1,1],[6,7],[10,15]	[13,15]	[1,1],[11,11]	–	[1,15]	–	[1,15]	[1,15]	–

Table 3: All intervals M, M' (number of visited peers) within the range [1,15] where performance of expanded queries is significantly better (M) or worse (M') than for the CORI baseline for Ohsumed in terms of MAP and for CiteSeer in terms of RP@10.

The picture for Ohsumed and GIRT (which is not shown here) is rather unclear. However, there are rather few cases in which expanded queries are actually significantly better than unexpanded ones. We also see that – when we concentrate on the very first peer in the ranking – virtually all query expansion methods fail to produce better results than the CORI baseline, in fact most of them are significantly worse. For Citeseer, the situation is easier to interpret: global pseudo feedback improves over unexpanded queries, but the other two expansion strategies are detrimental in all ranges.

All in all, query expansion seems a dangerous – instead of a helpful – tool when applied to ranking peers w.r.t. queries, regardless of whether we use global (from the web) or local information in the expansion process.

5.4 Profile adaptation

We now turn to the evaluation of the profile adaptation technique presented in section 3.4 and described by formula 4.

The first 702,892 queries of the original Citeseer query log were used for training and the retrieval with adapted profiles was then performed on the usual test set, consisting of the last 10,000 queries of the log. Updates of profiles were only performed during training, not during the evaluation of queries in the test set. From now on, all results are in terms of RP@10 only since we do not have relevance judgments for Citeseer.

Figure 2 (a) shows the performance of CORI baseline runs that use adapted profiles as compared to the CORI baseline with unadapted profiles for profile lengths of 10 and 80 terms. There is improvement for each number of peers visited. Figure 2 (b) shows the relative improvement of adapted profiles over unadapted ones as a function of the number of peers visited, calculated as

$$\frac{RP@10_{adapted} - RP@10_{base}}{RP@10_{base}} \tag{7}$$

where $RP@10_{adapted}$ and $RP@10_{base}$ refer to the RP@10 scores of the adapted run and the corresponding CORI baseline run, respectively.

We can see that generally the relative improvement of profile-adapted runs over the baseline is always greater than 5%, i.e. can be considered to be meaningful. Apart from profile length 10 the relative improvement curves have very similar shapes, with a tendency of smaller profiles gaining more from profile adaptation and relative improvement decreasing as more peers are visited.

Among the first 15 peers, the relative improvement for all profile sizes is greater than 10%, the improvement for the very first peer is at around 20% for all profile sizes except 10 terms.

All in all, the results of profile adaptation experiments are very encouraging. Of course, one should note that this analysis only revealed the *potential* improvement gained by profile adaptation: the evaluation optimistically assumed that training queries reach all peers that possess one of the query terms, something which is probably not the case in a P2PIR system. However, if a query routing algorithm maintains a random component (so that peers that are initially ranked lowly for a query may still be visited), this potential is very likely to be exploited – although it may take more queries to do so.

6. CONCLUSIONS

In this paper, we have studied the problem of resource selection in distributed text retrieval systems, i.e. the task of ranking a set of information resources w.r.t. a given user query and then employing the ranking for choosing a subset of resources to retrieve documents from. More precisely, it was investigated how many terms can be pruned from profiles without degrading performance significantly and what techniques can improve peer rankings.

In summary, the results of this paper suggest that it is possible to prune profiles substantially without losing retrieval effectiveness. They also show a good potential of profile adaptation techniques, as opposed to query expansion that did not improve results.

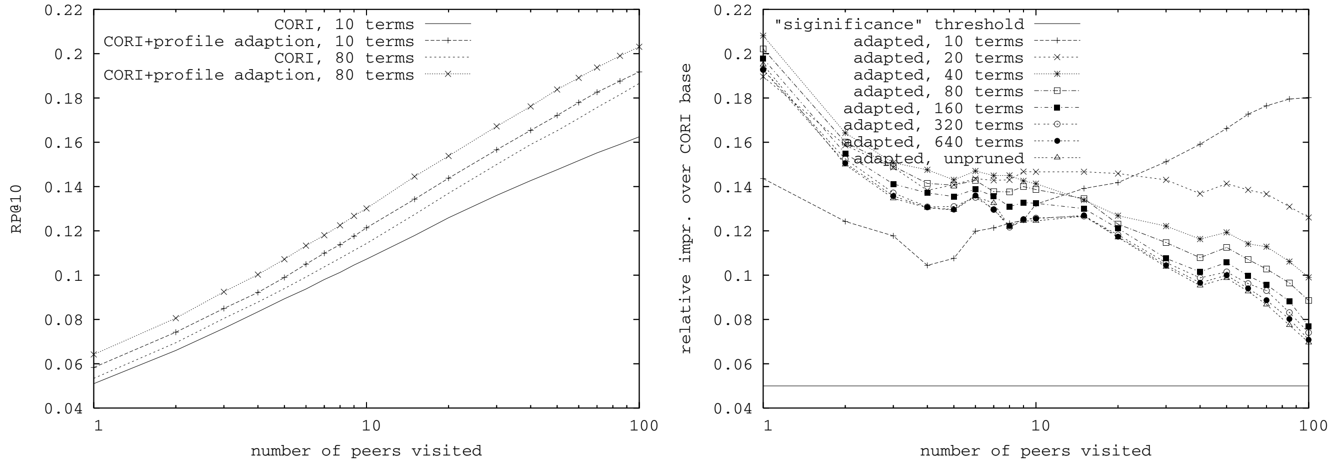
7. ACKNOWLEDGEMENTS

I would like to thank Lee Giles for providing the CiteSeer query log to me and for his prompt reactions to all my enquiries about it. Special thanks also go to Florian Holz, Gregor Heinrich and Sven Teresniak for the excellent discussions on relative precision; and to Norbert Fuhr who pointed me into the right direction regarding its final formulation.

8. REFERENCES

[1] R. Akavipat, L.-S. Wu, F. Menczer, and A.G. Maguitman. Emerging semantic communities in peer web search. In *P2PIR '06: Proceedings of the International Workshop on Information Retrieval in Peer-to-Peer Networks*, pages 1–8, 2006.

[2] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in P2P search engines. In *Proceedings of SIGIR '05*, pages 67–74, 2005.



(a) (b)

Figure 2: Performance of runs with adapted profiles as a function of the number of peers visited in terms of (a) RP@10 – where performance of the CORI baseline is given for comparison and (b) relative improvement over the CORI baseline.

[3] M. Bender, S. Michel, and G. Weikum. P2P Directories for Distributed Web Search: From Each According to His Ability, to Each According to His Needs. In *Proc. of ICDEW '06*, page 51, 2006.

[4] J. Broekstra, M. Ehrig, P. Haase, F. van Harmelen, M. Menken, P. Mika, B. Schnizler, and R. Siebes. Bibster - A Semantics-Based Bibliographic Peer-to-Peer System. In *Proceedings of SemPGRID '04, 2nd Workshop on Semantics in Peer-to-Peer and Grid Computing*, pages 3–22, 2004.

[5] J. Callan. Distributed Information Retrieval. In W.B. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

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

[7] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of SIGIR '95*, pages 21–28, 1995.

[8] S. Chernov, P. Serdyukov, M. Bender, S. Michel, G. Weikum, and C. Zimmer. Database Selection and Result Merging in P2P Web Search. In *Third International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005)*, 2005.

[9] 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 *12th International Symposium on High Performance Distributed Computing (HPDC)*, 2003.

[10] J. C. French, A. L. Powell, J. Callan, C. L. Viles, T. Emmitt, K. J. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In *Proceedings of SIGIR '99*, pages 238–245, 1999.

[11] Gnutella. The Gnutella Protocol Specification v0.4. Available from www9.limewire.com/developer/gnutella_protocol_0.4.pdf, 2001.

[12] L. Gravano, H. García-Molina, and A. Tomasic. GlOSS: text-source discovery over the Internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.

[13] D. Hawking and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems*, 17(1):40–76, 1999.

[14] S. Joseph. Neurogrid: Semantically routing queries in peer-to-peer networks. In *Proceedings of the International Workshop on Peer-to-Peer Computing*, 2002.

[15] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proceedings of the Conference on Information and Knowledge Management*, pages 300–307, 2002.

[16] A. Z. Kronfol. FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine, 2002.

[17] A. Loeser, S. Staab, and C. Tempich. Semantic Social Overlay Networks. *IEEE JSAC - Journal on Selected Areas in Communication*, 25(1):5–14, 2007.

[18] J. Lu and J. Callan. Pruning long documents for distributed information retrieval. In *Proceedings of CIKM'02*, pages 332–339, 2002.

[19] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 199–206, 2003.

[20] P. Ogilvie and J. Callan. The effectiveness of query expansion for distributed information retrieval. In *Proceedings of CIKM'01*, pages 183–190, 2001.

[21] D. Puppin, F. Silvestri, and D. Laforenza. Query-driven document partitioning and collection selection. In *Proceedings of InfoScale '06*, pages 34–41, 2006.

[22] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC-3. In *Proceedings of TREC-3*, pages 21–30, 1992.

[23] S. Saroiu, P. Gummadi, and S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of Multimedia Computing and Networking*, 2002.

[24] M. Shokouhi and J. Zobel. Federated text retrieval from uncooperative overlapped collections. In *Proceedings of SIGIR '07*, pages 495–502, 2007.

[25] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proceedings of SIGIR '95*, pages 172–179, 1995.

[26] H.F. Witschel. Global Term Weights in Distributed Environments. *Information Processing and Management*, 44(3):1049–1061, 2008.

[27] H.F. Witschel, F. Holz, G. Heinrich, and S. Teresniak. An Evaluation Measure for Distributed Information Retrieval Systems. In *Proceedings of ECIR'08*, 2008.

[28] L.-S. Wu, R. Akavipat, and F. Menczer. 6S: Distributing crawling and searching across Web peers. In *Proceedings of WTAS2005*, 2005.

[29] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of SIGIR '98*, pages 112–120, 1998.

[30] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of SIGIR'96*, pages 4–11, 1996.