Aggregation of Document Frequencies in Unstructured P2P Networks

Robert Neumayer, Christos Doulkeridis[*], and Kjetil Nørvåg

Norwegian University of Science and Technology
Sem sælands vei 7-9, 7491, Trondheim, Norway
{neumayer,cdoulk,noervaag}@idi.ntnu.no

Abstract. Peer-to-peer (P2P) systems have been recently proposed for providing search and information retrieval facilities over distributed data sources, including web data. Terms and their document frequencies are the main building blocks of retrieval and as such need to be computed, aggregated, and distributed throughout the system. This is a tedious task, as the local view of each peer may not reflect the global document collection, due to skewed document distributions. Moreover, central assembly of the total information is not feasible, due to the prohibitive cost of storage and maintenance, and also because of issues related to digital rights management. In this paper, we propose an efficient approach for aggregating the document frequencies of carefully selected terms based on a hierarchical overlay network. To this end, we examine unsupervised feature selection techniques at the individual peer level, in order to identify only a limited set of the most important terms for aggregation. We provide a theoretical analysis to compute the cost of our approach, and we conduct experiments on two document collections, in order to measure the quality of the aggregated document frequencies.

1 Introduction

Modern applications are often deployed over widely distributed data sources and each of them stores vast amounts of data, a development partly driven by the growth of the web itself. Web information retrieval settings are a good example for such architectures, as they contain large document collections stored at disparate locations. Central assembly of the total information is neither feasible, as digital rights do not allow replication of documents, nor effective, since the cost of storing and maintaining this information is excessive. In order to achieve interoperability and intercommunication, there exists a need for loosely-coupled architectures that facilitate searching over the complete information available. Peer-to-peer (P2P) networks constitute a scalable solution for managing highly distributed document collections and such systems have often been used in web information retrieval and web search settings [3, 6, 7, 18, 15].

[*] This work was carried out during the tenure of an ERCIM ”Alain Bensoussan” Fellowship Programme.

One of the main problems in distributed retrieval lies in the difficulty of providing a qualitative ranking of documents, having as reference the centralised case. At the same time, performance and scalability considerations play a vital role in the development and applicability of such a widely distributed system. Thus, the important problem in the context of unstructured P2P networks is to provide a comprehensive ranking of terms (and documents). Clearly, exchanging all terms and their respective document frequencies would be a solution, however the cost is prohibitive, even for modest network sizes and medium-sized document collections. Therefore, we need a pre-selection of terms at peer level to evaluate the usefulness of terms locally, in order to decide which ones shall be aggregated. The usage of only a sub-part of all terms of a peer is further motivated by the possibility of holding back information, i.e. the more flexible an approach handles such terms, the more stable it is with respect to these types of inaccuracies. This process must work well without consuming excessive bandwidth, regardless of the size of the network topology. Also, the process should not be too specific with respect to the single collections, as both the distribution and the size of the local collections may vary significantly. These are the main issues to be investigated in this paper.

In our approach, peers first form a hierarchical overlay in a self-organising manner, which enables efficient aggregation of information. Then, carefully selected terms and their corresponding frequencies from each peer are pushed upwards in the hierarchy. At the intermediate levels, common terms from different peers are aggregated, thus reducing the total amount of information transferred. At the top levels of the hierarchy, a hash-based mechanism is employed to compute the global frequency values of terms, without requiring a single peer to perform this task. Finally, the information is disseminated to all peers and can be used for ranking documents.

Towards this goal, we investigate the impact of unsupervised feature selection techniques for term selection, i.e. techniques of selecting only the most useful terms of an often prohibitively large overall set of terms. Unsupervised refers to techniques which do not use available class information for term ranking as is often used in the machine learning context, if available. Such techniques can be applied on each peer autonomously, without explicit common assumptions, such as the availability of common labels, as in the case of supervised feature selection. Moreover, as the number both of documents and topics for each peer may vary, feature selection is an important tool, in order to identify terms that a peer is an expert on and can contribute to compute the correct document frequency value for.

The contributions of this work are: 1) we propose a hierarchical term aggregation method, which estimates global document frequencies of terms without assembling all information at a central location, suitable for unstructured P2P networks, 2) we investigate how unsupervised feature selection techniques applied at peer level affect the accuracy of the aggregated information, 3) we provide a cost model to assess the requirements of our approach in terms of transferred data, and 4) we conduct an experimental evaluation on two document collections, one of moderate size to show the applicability of our ideas, and one large collection of over 450.000 documents to demonstrate the scalability and application to web-based data.

The remainder of this paper is structured as follows: in Sect. 2, we provide an overview of the related work. We describe the aggregation process, starting from a description of the architecture, an overview of unsupervised feature selection methods employed at peer level, and eventually by presenting a cost model for assessing the communication cost, in Sect. 3. The experimental evaluation is presented in Sect. 4. Finally, in Sect. 5, we conclude the paper and give an outlook on future work.

2 Related Work

Distributed information retrieval (IR) has advanced to a mature research area dealing with querying multiple, geographically distributed information repositories. Both term weighting and normalisation are identified as major problems in dynamic scenarios [21], for both require global document frequency information. Viles and French study the impact of document allocation and collection-wide information in distributed archives [20]. They observe that even for a modest number of sites, dissemination of collection-wide information is necessary to maintain retrieval effectiveness, but that the amount of disseminated information can be relatively low. In a smaller scale distributed system, it is possible to use a dedicated server for collecting accurate term-level global statistics [10]. However, this approach is clearly not appropriate for large-scale systems.

In [22], the authors examine the estimation of global term weights (such as IDF) in information retrieval scenarios where a global view of the collection is not available. Two alternatives are studied: either sampling documents or using a reference corpus independent of the target retrieval collection. In addition, the possibility of pruning term lists based on frequency is evaluated. The results show that very good retrieval performance can be reached when just the most frequent terms of a collection (an extended stop word list) are known, and all terms which are not in that list are treated equally. The paper does not consider how to actually determine (collect) and distribute this information.

Moreover, we implicitly want to study the effects of pre-selection methods on overlay network generation. Also, our experiments are specifically designed to show the effects of unequally distributed partition sizes.

Content-based search in P2P networks [16] is usually related to full-text search [9, 19, 24], with most approaches relying on the use of structured P2P networks. Some research focuses on providing P2P web search functionalities, like in [11], where MINERVA is presented, a P2P web search engine that aims at scalability and efficiency. In MINERVA, each peer decides what fraction of the Web it will crawl and subsequently index. In further work, the authors also presented an information filtering approach relaxing the common hypothesis of subscribing to all information resources and allowing users to subscribe to the most relevant sources only [25]. Previous approaches regarding P2P web search have focused on building global inverted indices, as for example Odissea [18] and PlanetP [6]. In PlanetP, summaries of the peers' inverted indices are used to approximate TF-IDF. Inverse peer frequency (the number of peers containing the term) is used instead of IDF. It is questionable how this would scale in large P2P networks with dynamic contents, as also noted in [2]. In [4] superpeers are used to maintain DF for the connected peers. A similar approach is also used in [12]. Bender et al. [5] study global document frequency estimation in the context of P2P web search. The focus is on overlapping document collections, where the problem of counting duplicates is immense. Their system relies on the use of an underlying structured P2P network. A similar approach is described in [13], which is quite different from our setup that assumes an unstructured P2P architecture.

A major shortcoming of all these approaches is that their efficiency degrades with increasing query length and thus they are inappropriate for similarity search. Recently, approaches have been proposed that reduce the global indexing load by indexing carefully selected term combinations [17].

Furthermore, several papers propose using P2P networks in a digital library context [2, 7, 8, 14, 15]. In [3], a distributed indexing technique is presented for document retrieval in digital libraries. Podnar et al. [14] use highly discriminative keys for indexing important terms and their frequencies. In [15], the authors present *iClusterDL*, for digital libraries supported by P2P technology, where peers become members of semantic overlay networks (SONs).

3 Hierarchical Aggregation Based on Term Selection

In this section, we describe our approach for aggregating terms and their document frequencies, without central assembly of all data. We employ an unstructured P2P architecture and the overall aim is to provide estimates of frequency values that are as similar as possible to the centralised case, where all documents are available at a single location. We first provide an overview of the DESENT architecture. We then describe how aggregation is realised within our framework along with the feature selection methods we employ on a local level.

3.1 Architecture

DESENT. In order to create a hierarchical overlay network over a purely unstructured (Gnutella-like) P2P network, no matter its network distance, we employ a variant of DESENT [8]. The reasons for this choice are the completely *distributed* and *decentralised* creation of the hierarchy, its low creation cost and robustness. The most important details of the basic algorithm are described in the following; for more in-depth explanations we refer to [8]. The DESENT hierarchy can be used for building overlays for searching, but also for other purposes like aggregation of data or statistics about contents from participating peers – which is the way that DESENT is utilised in this paper.

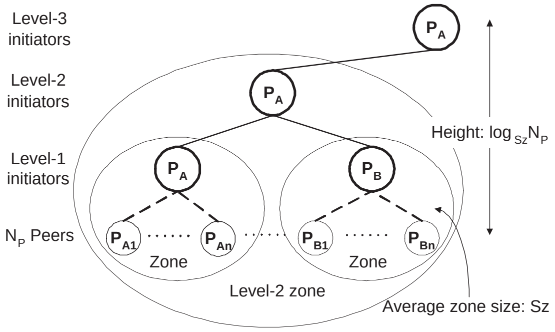


Fig. 1. Example of a P2P hierarchy of height h=3 with peers and zones

For an illustrative example of the DESENT hierarchy, see Fig. 1. The bottom level consists of the individual peers ($P_{A_1} \dots P_{A_n}$ and $P_{B_1} \dots P_{B_n}$). Then neighbouring peers (network-wise) create *zones* of approximate size S_Z peers (i.e. groups of peers) around an *initiator* peer (P_A and P_B), which acts as a zone controller. Notice that the height (h) of the hierarchy equals to: $log_{S_Z} N_P$. These level-1 initiators (P_A and P_B) are mostly uniformly distributed over the network, and are selected independently of each other in a pseudo-random way. The initiators form the next level of the hierarchy, they are responsible for the peers in their zones, and they aggregate the summary information of their peers into more abstract summaries.

In the subsequent phases, super-zones are created, which consist of a number of neighbouring zones from the previous level. Each super-zone is represented by a super-zone initiator that is responsible for the initiators in its zone and aggregates the information of these initiators. The zone initiators essentially form a P2P network similar to the original P2P network, and the aforementioned process is repeated recursively, using the zone initiators as peers. In the example of Fig. 1, P_A is initiator both at level-2 and level-3. In this way, a hierarchy of initiators is created, with each initiator creating summaries of information that refer to the contents of all peers in the tree rooted at that initiator. Finally, at the top-level initiator, summary information that spans the contents of the entire network is available.

Aggregation Process. The process of estimating the *frequency of selected terms* can be summarised as follows:

1. A tree-based P2P structure is created using the DESENT protocol [7, 8].
2. All peers select up to T terms from their local document collection using one of the techniques described in Sect. 3.2, and send these terms together with the total number of documents to the parent peer in the tree.
3. Each parent peer receives up to $S_Z T$ terms with respective document frequencies, where S_Z denotes the average number of peers in a zone. The

parent peer selects up to T terms, these terms are propagated upwards together with the aggregated document frequencies and the total number of documents in the subtree rooted at the peer.

4. The process continues up to the level of the children of the root (i.e., peers at level $h-1$), where h denotes the height of the tree. Level 0 is the bottom level and level h is the level of the root peer. Instead of performing the last aggregation at the root peer, it is performed by the children of the root. This is achieved by first distributing their aggregated values by hashing to the other root-children peers, and after processing these, the peers send all their aggregated results to all the other level $h-1$ peers.
5. The estimated document frequency values and the total number of documents are disseminated to the participating peers.
6. The whole process is repeated at regular intervals, in order to capture changes in document contents, as well as improving the estimated values. An alternative to fixed-time intervals would be to employ heuristics to assess the fluctuation in the network, i.e. initiate the process once a given number of peers joins or leaves the network.

Local Feature Selection and Document Frequency Calculation. Each peer P_i selects up to T terms from the $N_{l,i}$ locally stored documents, using one of the unsupervised feature selection techniques described in Sect. 3.2. Feature selection at a peer is based on the peer's local knowledge only. Thus, the result of the feature selection is a *term vector* TV_i, which is the number $N_{l,i}$ and vector of term tuples. Each term tuple in TV_i contains a term t_j and the local document frequency d_j: $TV_i = [N_{l,i}, [(t_1, d_1), ..., (t_T, d_T)]]$.

Level-wise Aggregation. After the $S_Z T$ selected terms from the previous phase have been received, a new term vector is created of the received terms and their frequencies, i.e., $TV_j = [N_s, [(t_1, d_1), ..., (t_{S_Z T}, d_{S_Z T})]]$. N_s is the sum of the received local frequencies, i.e., $N_s = \sum_{i=1}^{S_Z} N_{l,i}$. Furthermore, duplicate terms and their frequencies (i.e., the same term originating from several peers) are aggregated into one tuple, so that in general, in the end the number of terms in the new term vector is less than $S_Z T$. Finally, the term vector is reduced to only contain T terms. Term selection is performed based on the frequency of appearance, therefore terms that have high frequency are favoured. The intuition, which is also conformed by related work in [22], is that it is important to identify terms that are globally frequent and forward such terms to the top of the hierarchy. The generated term vector after aggregation and term selection, again consisting of T terms, is sent to the next level in the tree and this process continues iteratively up to level $h-1$, i.e., the children of the root.

Hash-based Distribution and Aggregation. Performing the final aggregation at the root peer is a straightforward process, however it makes the system vulnerable, as it induces a single point of failure. Instead, the final aggregation is performed by the children of the root, at level $h-1$. Notice that in this phase, our approach trades efficiency for robustness. We employ a more costly way to aggregate information, however the overall system becomes fault-tolerant. The actual aggregation is achieved by having the level $h-1$ peers first distributing their aggregated values, by hashing, to the other level $h-1$ peers. A recipient peer becomes responsible for a different subset of terms and aggregates their frequencies, thus performing (part of) the task that the root peer would perform. After the aggregation of the received term vectors, the peers send all their aggregated results to the other level $h-1$ peers. In the end, all level $h-1$ peers have the complete aggregated values locally available.

The reason for hashing is two-fold. First, it is important that all statistics for one particular term end up at the same node, in order to provide aggregated values per term. Second, the workload of the final aggregation is distributed and shared among the level $h-1$ peers, thus achieving load-balancing.

Dissemination of Information. In the final phase, the aggregated term vectors are distributed to all participating peers. This is performed by using the hierarchy as a broadcast tree. The term vectors are sent using the tree, until they reach the level-0 peers. The size of the disseminated information is equal to the number of term vectors ($S_Z T$) multiplied by the number of level $h-1$ peers. The aggregated terms and document frequencies are now available at all peers locally. As a consequence, any peer can use this information, in order to provide rankings of terms and documents taking into account the global document collection. In the experimental section, we study the accuracy of relevant ranking between pairs of terms to demonstrate the effectiveness of our approach.

3.2 Local Term Selection Approaches

Feature selection algorithms can generally be categorised as either supervised or non-supervised. Supervised methods use provided labels or class assignments for documents. The best features are then selected according to their class labels and the distribution of the feature across classes. In many cases, however, class labels are not available. In the context of distributed collections, such labels are particularly rarely available due to reasons of missing common document types or the general ad-hoc character of the collections themselves. To perform feature selection nevertheless, unsupervised techniques – even though they are fewer than supervised ones – can be used. These methods mainly rely on frequency information of a feature or term within a collection and judge its usefulness.

Following the vector space model of information retrieval we use N as the number of documents in a collection (which can be either global, i.e. the whole collection, or local when only a subset of the collection is considered). Further we use *df(t)* for the number of documents a term occurs in, also called the document frequency of term t. The number of occurrences of term t in document d is denoted to as the term frequency *tf(t,d)*. In this context, we propose the usage of the following unsupervised methods as possible local feature selection methods in the DESENT system.

Document Frequency (DF). One of the most prevalent techniques is denoted as document frequency thresholding. The main assumptions underlying document frequency thresholding are that terms occurring in very many documents carry less discriminative information and that terms occurring only in very few documents will provide a strong reduction in dimensionality (even though they might be discriminative in some cases). In combination with an upper and lower threshold, feature selection can be applied. This leads to results comparable to supervised techniques.

Collection Frequency (CF). The collection frequency of a term is given by the sum of all term frequencies for a given term (the total number of occurrences of a term in a collection):

$$cf(t) = \sum_{i=0}^{N} tf(t, d_i) \tag{1}$$

The collection frequency therefore ranks terms differently which occur only in few documents but with a higher term frequency.

Collection Frequency Inverse Document Frequency (CFIDF). The collection frequency inverse document frequency is represented by weighting the collection frequency values by the inverse document frequency for a term:

$$cfidf(t) = cf(t)log2(N/df(t)) \tag{2}$$

This measure can possibly cover both aspects the local document frequency and total number of occurrences for a term.

Term Frequency Document Frequency (TFDF). Another, quite recent technique to exploit both the *tf* and *df* factors is presented in [23]:

$$TFDF(t) = (n_1 n_2 + c(n_1 n_2 + n_2 n_3)) \tag{3}$$

n_1 denotes the number of documents in which t occurs, n_2 the number of documents t occurs only once, and $t3$ the number of documents containing t at least twice. An increasing weight c gives more weight for multiple occurrences.

Weirdness Factor (WF). The weirdness factor [1] was initially used to better distinguish special language text from rather common language use. The underlying idea is to identify terms which are very specific to a given collection. Terms have a high weirdness, i.e. are very specific to a given collection, if the ratio between relative local frequency and relative frequency in the reference collection is high:

$$weirdness(t) = \frac{\frac{cf_l(t)}{N_l}}{\frac{cf_r(t)}{N_r}} \tag{4}$$

Here, cf_l denotes the frequency of a term in the local collection, N_l the number of documents in the local collection; cf_r and N_r are the respective values for the reference corpus collection. In our case, we use the British national corpus as reference collection[1] which is a 100 million word corpus representing everyday English. This is feasible since all our collections are in English, otherwise reference corpora in other languages would be necessary.

3.3 Cost Analysis

We employ a simple cost analysis to assess the bandwidth consumption of the proposed approach. The basic parameters that influence the total communication cost (C_{total}) are: the number of peers (N_P) in the network, the average zone size (S_Z), the number of terms (T) in the term vectors propagated by each peer to its parent, and the size of the tuple representing each term (t_{size}). Each tuple of a term vector contains a term (we use as average size 16 characters for representation) and a frequency value (4 bytes). Hence, each tuple needs t_{size}=20 bytes. Moreover, each term vector is accompanied by a number (integer) that represents the number of documents associated with the term vector, however this cost is negligible compared to the size of the term vector. Notice that the height of the hierarchy (h) is derived as h=$log_{S_Z} N_P$.

The total number of terms (T_{up}) propagated upwards at each level is calculated by multiplying the number of peers (or initiators) at that level with the number of terms (T) per peer. Thus, the number of terms propagated up until the children of the root are given by:

$$T_{up} = \sum_{i=0}^{h-2} (\frac{N_P}{(S_Z)^i} T) = N_P T + \frac{N_P}{S_Z} T + \ldots + \frac{N_P}{(S_Z)^{h-2}} T \tag{5}$$

Thus, the cost for propagating term vectors upwards can be derived as:

$$C_{up} = T_{up} t_{size} = N_P T t_{size} \sum_{i=0}^{h-2} \frac{1}{(S_Z)^i} \tag{6}$$

There exists also a communication cost (C_h) related to hashing the information at the children of the root. Each child hashes its $S_Z T$ term vectors to the other children, and the number of children is $\frac{N_P}{(S_Z)^{h-1}}$, leading to cost

$$C_{out} = S_Z T t_{size} \frac{N_P}{(S_Z)^{h-1}} = T t_{size} \frac{N_P}{(S_Z)^{h-2}}$$

Then all children need to recollect the aggregated term vectors, leading to a cost $C_{in} = N_P (T t_{size} \frac{N_P}{(S_Z)^{h-2}})$. Consequently, the total cost is equal to:

$$C_{total} = C_h + C_{up} = (C_{in} + C_{out}) + C_{up} = N_P T t_{size} (\frac{N_P + 1}{(S_Z)^{h-2}} + \sum_{i=0}^{h-2} \frac{1}{(S_Z)^i}) \tag{7}$$

[1] http://www.natcorp.ox.ac.uk/

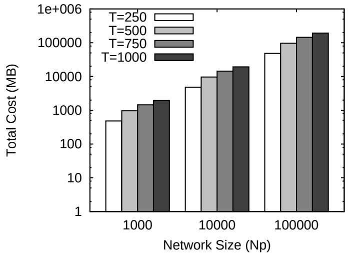


Fig. 2. Total cost (C_{total}) for hierarchical term aggregation

Obviously, compression techniques can further reduce the total cost, however this is out of the scope of this paper. Moreover, the cost for the creation of the DESENT hierarchy is described in [8] and it is not included in this analysis.

In Fig. 2, we graphically depict the total cost in MB for various networks sizes (N_P) ranging from 1K to 100K peers. We use varying values for T ranging from 250 to 1000 terms. Notice that the y-axis is in logarithmic scale. Notice that the total cost corresponds to approximately 1MB per peer, even for large network sizes. Moreover, the total cost is controlled by decreasing the T value.

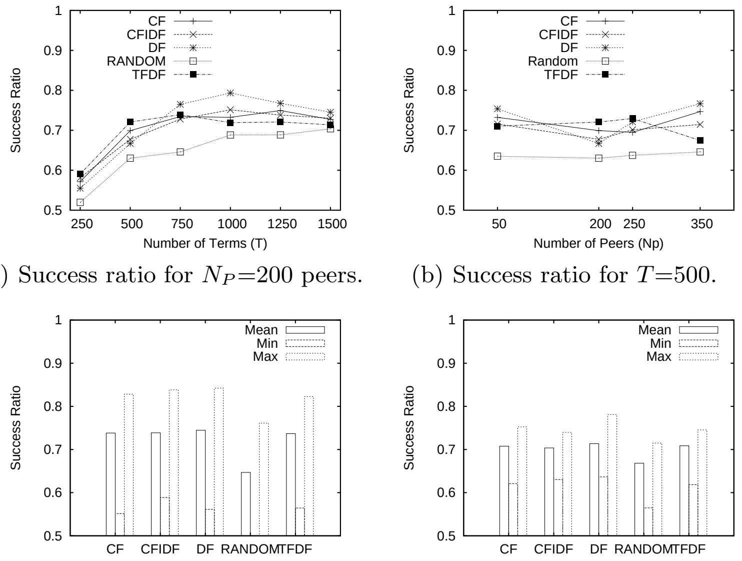
4 Experiments

We conducted experiments using two document collections. The 20 newsgroups data set[2] consists 18,828 newsgroup documents labelled by and (nearly) evenly distributed across 20 different classes (the groups the articles were posted to). The DMOZ collection is a collection of 483,000 English web pages, which are classified by the DMOZ taxonomy[3]. The collection has been created by retrieving the web pages that are linked from the leaf-classes of the DMOZ taxonomy. The taxonomy path to a page is considered to be the class/category of the page. Both test collections were preprocessed in terms of tokenizing, stop word removal and stemming for the English language.

We identify the following basic parameters for our experiments and study their effect. First, the *number of partitions* or peers, as it affects the scalability of our approach. Then, *the distribution factor*, defined as the size distribution of the local partitions. A high distribution factor denotes equal amounts of documents per partition. Last, the *document similarity*, defined as the degree to which documents in one partition are similar to each other. This simulates cases such as topically homogeneous collections (with a high degree of similarity) or cases of randomly distributed collections. To this end, we use class labels of documents and distribute documents to partitions already containing similar documents, with a higher or lower probability according to the setting. In the case where no

[2] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

[3] `http://www.dmoz.org`



(a) Success ratio for N_P=200 peers.

(b) Success ratio for T=500.

(c) Variance for N_P=50 peers.

(d) Variance for N_P=350 peers.

Fig. 3. Experimental results for the 20 newsgroups collection

labels are available, document clustering could be used instead to determine a measure of similarity.

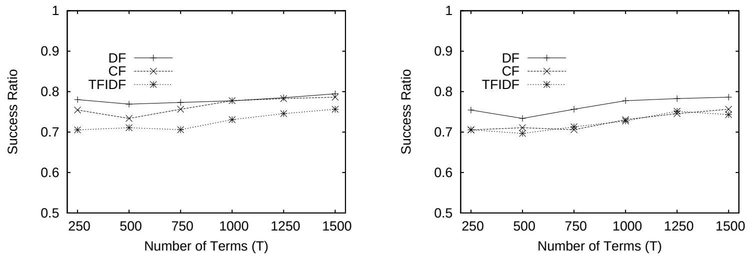
In our experimental evaluation we use varying settings, in order to simulate different use cases. We vary the number of peers to study the scalability of our approach. For each given number of peers, we apply four settings: 1) low similarity, high distribution, 2) low similarity, low distribution, 3) high similarity, high distribution, and 4) high similarity, low distribution. To be able to show the impact of all extreme values of both parameters, we also included mixed setups and also the case of documents which are distributed in equal sized partitions and have no similarity relation to each other at the other end of the spectrum. We apply the aforementioned feature selection methods at the local peer level and further added a random selection experiment to see the actual impact of the techniques with respect to no feature selection performed and to show the overall feasibility of the aggregation method.

4.1 Results for the 20 Newsgroups Collection

In Fig. 3, we study the quality of the aggregated document frequencies in terms of ranking. For this purpose, we define as *success ratio* the percentage of pairs of terms that have the same relative ranking in our approach and in the centralised case. In other words, for any two terms t_i and t_j the success ratio is the fraction of the number of such pairs with the same ranking with respect to the centralised ranking, over all possible combinations of pairs of terms. We chose this performance measure for existing approaches such as the Spearman or Kendall tau rank order correlation coefficients lack the support for rankings of different lengths, our approach, however, is closely related and basically extends these methods in its ability to handle different lengths of involved rankings

Fig. 3(a) shows the results for a network of N_P=200 peers for varying values of T. All feature selection methods achieve high values of success ratio, and the results improve with increasing T up to 1000 terms, since more information is propagated upwards and aggregated. For larger values of T, most methods exhibit a decrease in success ratio, due to more unimportant terms being aggregated thus causing noise, and this effect is stronger in small-size collections, such as 20 newsgroups. Notice that even the random selection achieves good performance, which is an argument in favour of the aggregation we employ – the propagated results are similar to the central case. Naturally, the intentional feature selection methods perform better by 10-15% except for the values obtained by the weirdness method which are omitted. In Fig. 3(b), we study the scalability with number of peers. We fix T=500 and the chart shows that the increased number of peers does not result in decreasing values of success ratio, an important finding for the scalability of our approach. Especially for small values of T the document frequency method is not the most stable one and the collection frequency methods provide better results. However, the document frequency performance increases with higher numbers of terms being aggregated.

In the following, we measure the mean values for the success ratio, along with minima and maxima. The values in Fig. 3(c) show the values for a total number of peers of N_P=50, while Fig. 3(d) shows the values for a total number of N_P=350. The standard deviation of results obtained by document frequency values for the smaller number of peers (Fig. 3(c)) is amongst the highest in this setup. When looking at both plots it is apparent that the $CFIDF$ is the more stable choice across different numbers of peers (and subsequently for higher numbers of documents per peer).



(a) Low similarity within peers. (b) High similarity within peers.

Fig. 4. Experimental results for the DMOZ collection. Results are for two experimental settings, low similarity within peers in 4(a), and high similarity in 4(b)

4.2 Results on DMOZ Collection

Fig. 4 shows experimental results using the DMOZ collection, in a network of N_P=784 peers. We provide the success ratio for a different number of terms analogously to Fig. 3. However, in this case the number of terms to be aggregated has a smaller impact on performance. This confirms our finding that the number of documents per peer strongly influences the overall result. We show results for high similarity within peers in Fig. 4(a) and low similarity in Fig. 4(b). The document frequency selection method performs best and the results across different numbers of terms to be aggregated are more stable than with the other collection. Again, this is due to the higher number of average documents per peer for this collection.

5 Conclusions and Future Work

In this paper, we proposed an efficient approach for aggregating the document frequencies of carefully selected terms in a loosely-coupled P2P network of digital libraries. We provided a cost model to assess the requirements of our approach in terms of communication, and we performed experiments on two document collections to demonstrate the impact of local feature selection on and the quality of the aggregated values. In our future work, we intend to study the results of ranking obtained by our approach, for document retrieval using keyword-based queries. Further, we plan on investigating techniques to handle different numbers of documents per peer as this proved to be the most difficult setting in our experiments. Also, we want to perform a more thorough evaluation on very large test collections.

References

1. K. Ahmad, L. Gillam, and L. Tostevin. Weirdness indexing for logical document extrapolation and retrieval WILDER. In *TREC*, 1999.
2. W.-T. Balke. Supporting information retrieval in peer-to-peer systems. In *Peer-to-Peer Systems and Applications*, 2005.
3. W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. DL meets P2P - Distributed document retrieval based on classification and content. In *Proc. of ECDL*, 2005.
4. W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. Progressive distributed top-k retrieval in peer-to-peer networks. In *Proc. of ICDE*, 2005.
5. 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.
6. F. Cuenca-Acuna, C. Peery, R. Martin, and T. Nguyen. PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. In *Proc. of HPDC*, 2003.
7. C. Doulkeridis, K. Nørvåg, and M. Vazirgiannis. Scalable semantic overlay generation for P2P-based digital libraries. In *Proc. of ECDL*, 2006.

8. C. Doulkeridis, K. Nørvåg, and M. Vazirgiannis. DESENT: Decentralized and distributed semantic overlay generation in P2P networks. *Journal on Selected Areas in Communications*, 25(1), 2007.
9. J. Lu and J. Callan. Full-text federated search of text-based digital libraries in peer-to-peer networks. *Information Retrieval*, 9(4), 2006.
10. S. Melink, S. Raghavan, B. Yang, and H. Garcia-Molina. Building a distributed full-text index for the web. *ACM Transactions on Information Systems*, 19(3), 2001.
11. S. Michel, P. Triantafillou, and G. Weikum. MINERVA infinity: A scalable efficient peer-to-peer search engine. In *Proc. of Middleware*, 2005.
12. H. Nottelmann and N. Fuhr. Comparing different architectures for query routing in peer-to-peer networks. In *Proc. of ECIR*, 2006.
13. O. Papapetrou, S. Michel, M. Bender, and G. Weikum. On the usage of global document occurrences in peer-to-peer information systems. In *Proc. of COOPIS*, 2005.
14. I. Podnar, T. Luu, M. Rajman, F. Klemm, and K. Aberer. A P2P architecture for information retrieval across digital library collections. In *Proc. of ECDL*, 2006.
15. P. Raftopoulou, E. G. M. Petrakis, C. Tryfonopoulos, and G. Weikum. Information retrieval and filtering over self-organising digital libraries. In *Proc. of ECDL*, 2008.
16. O. D. Sahin, F. Emekçi, D. Agrawal, and A. E. Abbadi. Contentbased similarity search over peer-to-peer systems. In *Proc. of DBISP2P*, 2004.
17. G. Skobeltsyn, T. Luu, I. P. Zarko, M. Rajman, and K. Aberer. Query-driven indexing for scalable peer-to-peer text retrieval. In *Proc. of Infoscale*, 2007.
18. T. Suel, C. Mathur, J. wen Wu, J. Zhang, A. Delis, Mehdi, X. L. Kharrazi, and K. Shanmugasundaram. Odissea: A peer-to-peer architecture for scalable web search and information retrieval. In *Proc. of WebDB*, 2003.
19. C. Tang and S. Dwarkadas. Hybrid global-local indexing for efficient peer-to-peer information retrieval. In *Proc. of NSDI*, 2004.
20. C. L. Viles and J. C. French. Dissemination of collection wide information in a distributed information retrieval system. In *Proc. of SIGIR*, 1995.
21. C. L. Viles and J. C. French. On the update of term weights in dynamic information retrieval systems. In *Proc. of CIKM*, 1995.
22. H. F. Witschel. Global term weights in distributed environments. *Information Processing and Management*, 44(3), 2008.
23. Y. Xu, B. Wang, J. Li, and H. Jing. An extended document frequency metric for feature selection in text categorization. In *Proc. of AIRS*, 2008.
24. J. Zhang and T. Suel. Efficient query evaluation on large textual collections in a peer-to-peer environment. In *Proc. of IEEE P2P*, 2005.
25. C. Zimmer, C. Tryfonopoulos, K. Berberich, M. Koubarakis, and G. Weikum. Approximate information filtering in peer-to-peer networks. In *Proc. of WISE*, 2008.