User Modeling for Full-Text Federated Search in Peer-to-Peer Networks

Jie Lu Jamie Callan

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

{jielu, callan}@cs.cmu.edu

ABSTRACT

User modeling for information retrieval has mostly been studied to improve the effectiveness of information access in centralized repositories. In this paper we explore user modeling in the context of full-text federated search in peer-to-peer networks. Our approach models a user's persistent, long-term interests based on past queries, and uses the model to improve search efficiency for future queries that represent interests similar to past queries. Our approach also enables queries representing a user's transient, ad-hoc interests to be automatically recognized so that search for these queries can rely on a relatively large search radius to avoid sacrificing effectiveness for efficiency. Experimental results demonstrate that our approach can significantly improve the efficiency of full-text federated search without degrading its accuracy. Furthermore, the proposed approach does not require a large amount of training data, and is robust to a range of parameter values.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process.

General Terms

Algorithms, Performance, Experimentation, Design

Keywords

User modeling, Query clustering, Peer-to-peer, Federated search

1. INTRODUCTION

User modeling for information retrieval has been a very active research topic in recent years. Most studies focus on using user models for query expansion/reformulation, result re-ranking, or document/link recommendation to improve the effectiveness of information access in centralized repositories. There have been very few studies of user modeling for federated search of multiple distributed collections in the absence of a central authority. Because peer-to-peer (P2P) networks have emerged as an attractive solution to information retrieval in distributed environments when search using centralized repositories or indexes is either impossible or impracticable, we are interested in investigating how user modeling can improve the performance of federated search in peer-to-peer networks.

Peer-to-peer networks contain three types of functional units (with each peer being a single functional unit or a combination of multiple functional units), namely *information providers* (search engines), *information consumers* (users), and *services* that provide functionality to facilitate efficient and effective search. One type of service that is essential to federated search in P2P networks is the *directory service*, which is responsible for routing queries and responses between consumers and providers. Hierarchical P2P architecture is a popular and effective peer-to-peer architecture used in large-scale operational and research systems. A hierarchical P2P network typically uses a two-level hierarchy of an upper-level of "*hubs*" for directory services and a lower-level of "*leaves*" for information providers and consumers. Each hub provides directory service to a region of the network. Multiple hubs work collectively to cover the whole network. Leaves can only connect to hubs. Hubs connect with leaves and other hubs.

We focus on *full-text* federated search in peer-to-peer networks, which conducts search over the full text of documents and returns results in relevance-based document rankings ("*full-text ranked retrieval*"). Full-text ranked retrieval aims to find content that satisfies the user's information need ("*informational search*"), which is in contrast to the simple Boolean retrieval commonly used for known-item search in file-sharing P2P networks. The process of full-text federated search in a hierarchical P2P network works as follows [9]. When a consumer has an information request, it issues a query to initially selected hub(s). A Time-To-Live (TTL) field in each query message determines the maximum number of times it may be relayed ("*search radius*"). A hub that receives the query uses its resource selection algorithm to rank and select one or more neighboring providers as well as hubs based on their full-text resource representations (typically the aggregations of the bag-of-words representations of individual documents) and routes the query to them ("*full-text provider/hub selection*"). A provider that receives the query uses its full-text document retrieval algorithm to generate a relevance-based ranking of its documents and responds with a list of the top-ranked documents. A hub is responsible for collecting the ranked lists returned by multiple providers, using its result merging algorithm to merge them into a single, integrated ranked list, and returning it to the consumer. Finally, a consumer merges results returned by multiple hubs.

The performance of federated search largely depends on how efficiently and effectively the information providers with relevant contents can be located ("*resource location*"). Because most current P2P networks provide very limited information to consumers about the available contents and their placement in the network, the resource location conducted by each consumer to initiate the search is typically no more than a random selection from a list of known directory services (hubs). Since there is no guarantee that the (arbitrarily) selected hub(s) can directly locate relevant resources, a relatively large search radius is usually required to reach the hubs that cover relevant contents. In this paper, we study how user modeling can be used to improve the

SIGIR'06, August 6–10, 2006, Seattle, Washington, USA.

quality of initial hub selection in order to start the search in the right neighborhood so that less effort (i.e., a smaller search radius) is required to find information providers with relevant contents and therefore higher search efficiency can be achieved. We propose an automatic mechanism for each information consumer to adaptively model the persistent, long-term interests of the user it represents based on past queries, and use the model to guide initial hub selection for future queries that represent interests similar to past queries. Our approach also automatically recognizes queries representing the user's transient, ad-hoc interests, for which the model based on search history cannot support effective initial hub selection, so it can fall back on a default search strategy of a large search radius to avoid sacrificing accuracy for efficiency. Experimental results demonstrate that our approach can significantly improve the efficiency of full-text federated search without degrading accuracy.

In the following section we describe related work on user modeling for information filtering, Web retrieval and federated search in P2P networks. Our approach to modeling user interests for full-text federated search is presented in Section 3, followed by the evaluation resources in Section 4. We present experimental results to evaluate the effectiveness of our approach in Section 5 and explore the impact of different parameter settings in Section 6. Section 7 concludes.

2. RELATED WORK

User modeling represents a user's information needs (interests) with a user model (also called "user profile"). A user model can be hand-crafted, or learned automatically from queries and their retrieved documents based on explicit (e.g., relevance judgment), implicit (e.g., mouse click, time on page), or pseudo (e.g., the top ranks) relevance feedback. Commonly used representations for a user model include a set of attributes with values, a structured representation such as a weighted concept hierarchy based on a predefined ontology, or a bag-of-words representation such as a vector of weighted terms or a unigram language model. A user's interests can be represented as a whole using one representation, or as a set of topics (determined by clustering or classification) with one representation for each topic. Which technique and representation to use in generating user models depends on the objectives and requirements of the particular applications.

In information filtering, a user model is specified by the user explicitly and modified over time by the system based on the long-term observations of the document stream and periodic relevance feedback from the user ("*adaptive information filtering*") [13]. Typically adaptive information filtering systems exploit machine learning techniques to handle positive and negative relevance feedback provided by the user. Although explicit relevance feedback from the user is usually available for information filtering, Web retrieval and federated search in distributed environments are unlikely to have such luxury.

Personalized Web search uses user models to adapt search to the needs of individuals through query expansion, result re-ranking, or link recommendation. A user model with a bag-of-words representation can be generated from queries and the text of the top-ranked or visited documents to represent the user's short-term interests (within a single search session) [15], long-term interests (over multiple search sessions) [19], or a combination of both [3, 18]. A user's long-term interests can also be modeled by classifying documents visited by the user into different ontology-based categories using complete document contents or snippets surrounding query terms [4, 17]. However, a predefined ontology and additional training data are required to learn the classifier.

Collaborative Web search assists a user in searching for information by utilizing others' expert knowledge or search experience. Typically query clustering is used to mine search engine query logs in modeling users' interests. To measure the similarity between queries for clustering, the overlap between query terms [21], the number of retrieved documents clicked in common [5, 21], the distance between the clicked documents based on their URLs or categories [1, 21], or a combination of the above [21] have been used. The contents of the clicked documents are not usually used, for efficiency reasons.

For federated search in peer-to-peer networks, there have also been a few attempts to utilize user interests to move a peer closer to those that more frequently provided relevant contents to its past queries in order to improve resource location for future queries [11, 14, 16]. The success of the approach relies on the existence of two properties in the network: i) similar contents are located near to each other ("*content-based locality*"), and ii) a consumer's queries are closely related to the persistent interests of the user it represents. Since contents relevant to a query tend to be similar, the first property guarantees that relevant contents are mostly near to each other and therefore resource location can be efficient. The network can provide this property by regulating its content placement using distributed hash tables [12] or dynamic topology evolution [2, 7, 10]. The search behaviors of information consumers generally exhibit the second property. However, since existing methods do not explicitly distinguish between queries that express the different interests of the user (e.g., sports versus music), they cannot tell which resources are more relevant to which interests, resulting in less efficient resource location when a resource relevant to one interest is selected to answer queries for other interests. Furthermore, because no method has been provided to explicitly separate transient information needs from those that are related conceptually to the user's persistent, long-term interests, the search performance of transient information needs (either efficiency or accuracy) is likely to be poor.

3. APPROACH DESCRIPTION

To eliminate the randomness of initial hub selection at each information consumer when initiating search in peer-to-peer networks, we propose to model the user's persistent, long-term interests based on past queries, and use the model to conduct initial hub selection for new queries according to the hubs' resource location effectiveness for old queries with similar interests ("*interest-based hub selection*"). Compared with existing methods of learning from search history [11, 14, 16], our approach distinguishes between different interests and measures the hubs' performance for each interest instead of modeling all interests as a single group. Our objective is that with effective initial hub selection to initiate search, a small search radius (and therefore little query routing) is sufficient to locate most relevant contents, improving search efficiency without sacrificing accuracy. The effectiveness of interest-based hub selection depends on whether the hubs capable of locating relevant contents efficiently and effectively for past queries perform well for future queries that express similar interests. A hierarchical P2P network in which the information providers having similar contents connect to the same hubs can best support effective interest-based hub selection since contents relevant to similar interests tend to be similar to each other. The network can provide this property by using distributed hash tables [12] or dynamic topology evolution [2, 7, 10] to regulate its content placement.

In addition to queries representing persistent information needs, a user may also issue queries that are not conceptually related to his/her long-term interests, to satisfy transient, ad-hoc information needs. Because interest-based hub selection at a consumer depends on the limited (and often biased) information the consumer has learned about the hubs as a byproduct of past search, it is unlikely to provide any clue about which hubs can best locate relevant contents for transient information needs not related to search history. Therefore, for queries expressing transient information needs, the consumer must resort to a more extensive search using a larger search radius (TTL) to route queries to the hubs that directly cover relevant contents, trading efficiency for accuracy. In other words, different search strategies are required for different types of queries to optimize the overall search performance. Our goal is to develop an approach that enables each consumer to distinguish between queries representing different persistent interests for effective interest-based initial hub selection at the consumer, and to recognize transient information needs so that full-text hub selection at the hubs can be fully utilized to guarantee accuracy.

User modeling for full-text federated search in a peer-to-peer network takes place at each individual information consumer due to the lack of a centralized server to monitor search activities in the network. Similar to the approach taken in [20], query clustering is used to group past queries in identifying a user's different interests. Each query cluster represents a *topic of interest*. The interest-dependent performance is measured for each hub that provided search results to this consumer, which is dynamically updated whenever new results are available. For a hub that covers contents related to multiple topics of interest, its performance for each topic is measured independently of the other topics. The optimal hubs for a new query are determined based on their performance for clusters of similar past queries.

In the following subsection, we present the design for the two main tasks of our approach, namely clustering queries and learning about the hubs' performance for each cluster. Section 3.2 describes in detail the implementation of our approach.

3.1 Design

Query clustering requires a representation for each query/cluster, and a similarity measure between queries and clusters. Because the small number of query terms does not provide a reliable basis for clustering queries effectively, a commonly used method to measure query similarity in Web retrieval is to count the number of commonly retrieved documents for the queries [5, 21]. This method may work well if the task is to group queries that are very similar. However, to group queries by interest, it is quite likely that two queries that express similar interests in a general topic (e.g., music) may not have any retrieved document in common even though the vocabularies of their retrieved documents may have significant overlap. Therefore, it is more appropriate to measure query similarity based on the *contents* of the documents returned for each query. Which retrieved documents to choose in generating a representation for the corresponding query depends on whether and what type of feedback is available. With explicit relevance feedback from the user, documents relevant to the query are selected. When feedback is implicit in the form of mouse clicks, the clicked documents are treated as relevant documents. The top-ranked merged documents are chosen in the last resort when neither explicit nor implicit feedback is available. After stopwords are removed and stemming is conducted, the contents of the chosen documents are used to generate a maximum likelihood unigram language model to represent the corresponding query. The representation of a query cluster is the aggregation of its members' language models. The similarity between a query and a cluster is measured by the Kullback-Leibler divergence between their representations.

Our choice of the clustering algorithm is guided by several characteristics of query clustering in peer-to-peer networks. First, because the sets of queries used for clustering are highly dynamic, the clustering algorithm should be incremental. Second, since the size of the query log at each individual information consumer is much smaller compared with the query logs of Web search engines, the clustering algorithm should be able to work well with limited data. Third, the algorithm should not require the number of clusters or the maximum size of each cluster to be set manually as it is unreasonable to assume that these parameters can be determined in advance. Based on the above considerations, we use a greedy non-hierarchical clustering algorithm that incrementally updates existing clusters to include new queries when their representations are similar to the old ones, or creates new clusters when they are sufficiently different in order to capture the user's new interests. Neither the number of clusters nor the size of each cluster is predetermined.

In previous research on using search history to improve federated search performance in P2P networks [11, 14, 16], search performance is measured by the number of documents returned for each query. For the known-item search that is common in P2P networks sharing music, videos, and software, this appears to be an appropriate measure since typically the search either returns relevant documents or returns no document at all. In contrast, full-text federated search is very likely to return non-relevant documents, so the number of documents returned is no longer a good measure of search performance. Because the top-ranked documents are more likely to be relevant than most lower-ranked documents, when no feedback is available, the information about how many documents returned by a hub appear among the overall top-ranked merged documents at a consumer is a more reliable indicator of the hub's performance for a query. Therefore, our approach uses this information as a surrogate for relevance feedback to measure each hub's performance on resource location for interest-based hub selection.

3.2 Implementation

Figure 3.1 shows an algorithmic description of our approach to user modeling for full-text federated search. Below we discuss several details that are important to its effectiveness.

When a query is issued, its query terms are used as its representation in determining which existing query clusters it is most similar to ("*classification*") for interest-based initial hub selection. However, the chosen query clusters are not necessarily the clusters that the query should join because we need a more reliable representation of the information need for effective query clustering. Therefore, incremental query clustering is conducted after the search results are obtained for the query so that the full contents of the top-ranked merged documents (assuming no feedback is available) can be used to generate the query representation for clustering. Our experimental results demonstrate that a small number of the top-ranked merged documents (e.g., 5–10) are sufficient to provide a reliable representation for query clustering. To distinguish between the two representations of the same query, we refer to the former one used for classification as its "*TermRepresentation*" and the latter one used for clustering as its "*DocRepresentation*".

We refer to queries that are related conceptually to the user's persistent interests as "*characteristic queries*" since they are characteristic of his/her long-term information needs. By contrast, queries that represent transient, ad-hoc interests of the user are referred to as "*uncharacteristic queries*". User modeling allows the consumer to use past search experience to reduce the search radius and improve search efficiency for characteristic queries without reducing accuracy. But a default, larger search radius is still required for uncharacteristic queries to reach more hubs so as to locate sufficient relevant contents. Therefore, for each new query, the consumer needs a classification threshold $T_{classify}$ to distinguish uncharacteristic queries from characteristic ones in order to apply different search strategies accordingly.

Each query cluster is required to reach a certain size S_{min} before it is regarded as a topic of interest. This is designed to avoid classifying queries to clusters of uncharacteristic queries formed by chance and to make the description of the topic represented by each cluster more reliable.

Instead of classifying a new characteristic query to the most similar cluster, a weighted *k*-nearest neighbor approach is used to increase the robustness of our approach, where the value of *k* is determined by $T_{classify}$ and the weight is related to the similarity between the query and the cluster.

Among all the clusters whose K-L divergence-based distance measures to a query's representation are small enough (determined by a clustering threshold $T_{cluster}$), the query chooses to join the largest cluster in order to minimize the "noise" introduced by small clusters of uncharacteristic queries.

The total number of query clusters can be limited in order to control the amount of resources dedicated by an information consumer to process and store the language models used to represent the clusters. Although in most cases a consumer may not find it necessary to limit the number of query clusters (the average size of the representation for a query cluster is 69KB in our experiments), associating each cluster with a time stamp and removing infrequently used clusters can reduce clusters of uncharacteristic queries and effectively model the user's interest shift. In our implementation, when the number of query clusters exceeds N_{max}, clusters among the *r* least recently used clusters are removed in an ascending order of cluster size until the number of query clusters drops to N_{max}.

4. EVALUATION RESOURCES

The hierarchical peer-to-peer network we used to evaluate the performance of our approach to full-text federated search was created from the data defined in a previously published P2P testbed [8]. 2,500 collections, each consisting of documents crawled from a real Web site, were extracted from the TREC WT10g Web test collection to define 2,500 information providers in a hierarchical P2P network. The number of hubs in the network was 50. The topology of the network was created to exhibit content-based locality (i.e., the information providers that directly connected to the same hub formed a cohesive content-based cluster). Each hub (directory service) served an average of 136 providers. The average shortest path length between any two hubs was 3.11. Full-text federated search used the mechanism briefly described in Section 1 and detailed in [9, 10].

```
PROCESSQUERY($q$)
  /* Compare new query to existing query clusters */
  $characteristic = false$
  initialize $M[\bullet] = 0$
  $q_t$ = TermRepresentation($q$)
  for each cluster $c_i$
     if KL($c_i$, $q_t$)<$T_{classify}$ AND $|c_i| \geq S_{min}$
        $characteristic = true$
        UpdateTimeStamp($c_i$)
        for each hub $h_j$ recorded by cluster $c_i$
              $M[h_j]$ += $NumTopDocs[c_i][h_j]/|c_i| \times \exp(-\mathrm{KL}(c_i, q_t))$
        end
     end
  end
  /* Classify new query as characteristic or uncharacteristic
  for retrieval */
  if $characteristic$
     SetTimeToLive($q$, $ttl_{characteristic}$)
     Sort hubs by $M[\bullet]$
     send $q$ to the $m$ top-ranked hubs
  else
     SetTimeToLive($q$, $ttl_{uncharacteristic}$)
     send $q$ to randomly selected $m$ hubs
  end
  /* Update query clusters with results for new query */
  get a set $R$ of the $D_{top}$ top-ranked merged documents for $q$
  initialize $N[\bullet] = 0$
  $q_d$ = DocRepresentation($R$)
  for each document $d_j$ in $R$
     $h_j$ = GetSourceHub($d_j$);
     $N[h_j]$++
  end
  if exists at least one cluster $c_i$ such that KL($c_i$, $q_d$)<$T_{cluster}$
     find the largest cluster $c$ among all $c_i$ with KL($c_i$, $q_d$)<$T_{cluster}$
  else
     $c$ = NEWCLUSTER()
     initialize $NumTopDocs[c][\bullet] = 0$
  end
  add $q$ to cluster $c$
  UpdateTimeStamp($c$)
  for each hub $h_j$ that responds to $q$
     $NumTopDocs[c][h_j]$ += $N[h_j]$
  end

NEWCLUSTER()
  if the total number of clusters = = $N_{max}$
     sort clusters by their time stamps
     delete the smallest cluster among the $r$ least recently used clusters
  end
  return new cluster
```

Figure 3.1 An algorithmic description of learning and using the user model at an information consumer for a query *q*

Two sets of queries were selected from the queries defined in the P2P testbed, which were automatically generated by extracting key terms from the documents in WT10g [8]. The first query set consisted of 563 characteristic queries manually chosen to

Table 4.1 "Broad" categories and stemmed sample queries.

Category	# queries	Sample queries
Music	72	Billy Joel Adam Ant album Jesse Jones play band
Financial information	67	capital Macquire common share Chrysler mortgage market product
Education	80	elementary educate Stanford university program District Columbia university college
Health	78	medical rehabilitate home care nurse primary care physician Santara
Technology	75	BSDI Internet free agent software secure product kerbero
Law	64	supreme court law resource legal federal war crime international law
Religion	67	lord Samuel Israel holy spirit testament god homosexual Jesus church sin
Government issues	60	tax cut budget deficit govern federal govern department
Uncharacteristic queries	437	Ocean Spray CraftWEB bookstore Torreblanca resort Acapulco

represent a user's persistent interests in 8 relatively broad categories, and 437 uncharacteristic queries automatically selected to express the user's transient information needs not related to the aforementioned 8 categories. The categories were determined by soft-clustering the 2,500 providers using their full-text resource representations and inspecting the most frequent non-stopword terms from each cluster. Therefore, these categories were representative of the contents provided in the network. Table 4.1 shows for each "broad" category a general description, the number of queries issued for the category, and sample queries with query terms stemmed using the k-stem stemmer [6]. Among the 8 categories, "Financial information", "Education", "Health" and "Technology" were popular in the network with a large number of providers providing related contents. By comparison, "Music", "Law", "Religion", and "Government issues" were much less popular. Samples of uncharacteristic queries are also included in the table. The second query set included 400 characteristic queries in 8 categories which can be regarded as sub-categories of the above "broad" categories and 600 uncharacteristic queries. Table 4.2 shows sample queries for these "narrow" categories.

Given a query set, queries in the set were issued by an information consumer in a random order. The information consumer was *not* given information about which queries represented which types of interests. It was up to the consumer to decide, based on the learned user model, whether to issue a query as a characteristic query (with interest-based initial hub selection and a small search radius) or as an uncharacteristic query (with a default large search radius). Because a small percentage of queries (depending on S_{min}) would be issued as uncharacteristic queries at the beginning in order to learn a reliable user model (even if the consumer recognized some of them as characteristic queries), the percentage of the queries issued as characteristic queries was expected to be slightly smaller than the percentage of the queries that were actually characteristic queries. For example, when S_{min} was 5, the percentage difference between them was 4%.

Table 4.2 "Narrow" categories and stemmed sample queries.

Category	# queries	Sample queries
Classical music	50	Bach sonata Richard Strauss record piano ninth symphony choral Beethoven
Stock information	50	stock split Dow Jones index Alcoa pay bonus dividend
Online education	50	distance educate enroll online course university phoenix online
Personal health	50	nutrition vitamin calorie fat pretzel fat oil cholesterol
Image processing	50	Adobe Photoshop image browse Kudo Epson photo image software
Civic regulation	50	water hazard rule waste pollution control sewage sludge regulate
Religious study	50	Christian theology religion history study lecture Islamic Muslim
Tax issues	50	income tax tax reform tax cut legislate

The 50 top-ranked documents returned by search using a single large collection consisting of all the contents of the 2,500 providers ("*single collection*" baseline) were treated as relevant documents for each query. The mean of the average precision over document cutoffs 1–30 was used to measure the accuracy of federated search over a set of queries. We refer to this measure as "*mean average overlap precision*" because it essentially measured the percentage of overlap between the documents returned by centralized search and those by federated search in the P2P network, i.e., the ability of a P2P network to mimic a good centralized search engine. Previous research has demonstrated that the automatically-generated queries and the "single collection" baseline are useful resources in studying federated search in P2P networks [8, 9, 10].

5. APPROACH EFFECTIVENESS

In this section we present evaluation results comparing the performance of full-text federated search using our method of interest-based initial hub selection based on user modeling against that using other methods of initial hub selection. Table 5.1 lists the parameter values we used for our approach.

Figure 5.1 shows the accuracy of full-text federated search (y-axis) using interest-based initial hub selection with query clustering when different percentages of the hubs could be located within the *search scope* (x-axis) for queries issued as characteristic queries. The search scope was determined by the number of hubs directly contacted by the consumer to issue queries (m, which varied from 1 to 5) and the maximum number of hops each query was allowed to be relayed (search radius) among the hubs in the network ($ttl_{characteristic}$, which varied from 0 to 3). The accuracy was measured by the mean average overlap precision over the set of queries that were issued as characteristic queries.[1] The performance of full-text federated search over the

[1] The accuracy of the set of queries issued as uncharacteristic queries was not included because they used $ttl_{uncharacteristic}$ and random initial hub selection.

Table 5.1 Parameters required by our approach.

Parameter	Description	Value
$T_{classify}$	The classification threshold	7.0
$T_{cluster}$	The clustering threshold	1.5
S_{min}	The minimum size of a cluster to represent a topic	5
D_{top}	The number of the top-ranked merged documents as pseudo-relevant documents	10
N_{max}	The maximum number of recorded query clusters	50
r	The number of least recently used clusters considered for removal when N_{max} is reached	$N_{max}/4$
$ttl_{uncharacteristic}$	Hub routing Time-To-Live for queries issued as uncharacteristic queries	4
$ttl_{characteristic}$	Hub routing Time-To-Live for queries issued as characteristic queries	0–3
m	The number of the hubs contacted by a consumer for a query	1–5

same set of queries by using random initial hub selection without user modeling was used as a baseline.

In our experiments, when less than 10% of the hubs were located within the search scope, no hub routing was involved so that federated search completely relied on initial hub selection to reach the hubs. Figure 5.1 shows that there was a big difference in accuracy between interest-based initial hub selection and random initial hub selection. This indicates that by modeling user interests, initial hub selection based on the interest each query represented was much more effective in choosing the hubs that cover most relevant contents. Overall, search started with random initial hub selection needed to rely on a much larger search scope and full-text hub selection for query routing among the hubs in order to obtain accuracy comparable to that started with interest-based initial hub selection. In other words, search based on the user model required a much smaller number of query messages and thus a much higher efficiency in order to achieve similar accuracy. In fact, when $ttl_{characteristic}$ was set to 0, the amount of query routing could be reduced by 80% compared with using a $ttl_{characteristic}$ of 3 (with m = 5), but relative degradation in accuracy was only 8%. In summary, the results demonstrate that our approach can significantly improve the efficiency of full-text federated search without degrading its accuracy.

One might expect that user interests that were more focused could be better modeled and could enable more effective interest-based hub selection. The results show that interest-based hub selection only had slightly superior results for queries of "narrow" categories. This can be explained by the fact that with the small number of hubs in the network, the content area covered by each hub was relatively broad, so interest-based hub selection was likely to select the same hubs for queries of a "broad" category and for queries of the corresponding "narrow" category, leading to similar performance. A finer granularity in the hubs' content areas might yield a greater performance improvement for queries of more focused interests.

In addition to random initial hub selection, we also compared our approach against three other methods of initial hub selection. *Content-based initial hub selection* uses resource selection algorithm based on the content models of the hubs learned from previous search results for initial hub selection. Each consumer cumulatively constructs its own hub models using the contents of the top-ranked documents it received from the hubs for previous queries. *Performance-based initial hub selection* selects hubs to initiate search based on their performance measured by the total number of documents returned by each hub that appear among the 10 top-ranked merged documents at the consumer for its previous queries. Both methods use the information from a user's search history for initial hub selection but do *not* explicitly construct a user model. Therefore, they do not have the ability to distinguish between characteristic queries representing persistent user interests and uncharacteristic queries expressing transient information needs, which means that only a single search strategy can be applied to all the queries. *Non-clustered interest-based initial hub selection* generates a non-clustered user model by aggregating previous queries and the contents of their 10 top-ranked retrieved documents, and uses the same performance measure as performance-based initial hub selection to evaluate the hubs. Because it constructs a user model explicitly, it has the potential to separate uncharacteristic queries from characteristic ones. However, it can only measure each hub's performance for past queries as a whole without distinguishing between the differences in performance for different interests.

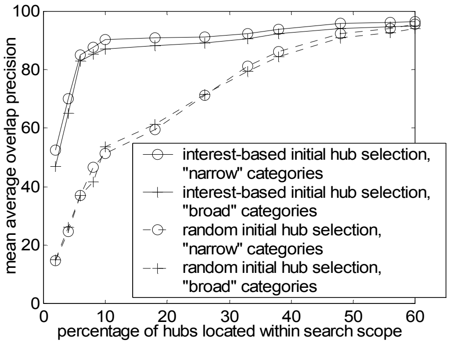

Figure 5.1 Search accuracy within different search scopes for interest-based initial hub selection.

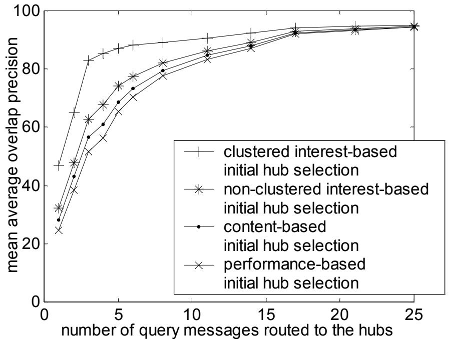

Figure 5.2 Search accuracy vs. search efficiency for different methods of initial hub selection.

Figure 5.2 plots the search accuracy (y-axis) against the number of query messages routed to the hubs (x-axis) for search using different initial hub selection methods. The larger the number of query messages, the lower the efficiency. The accuracy for an interest-based method (clustered or non-clustered) was measured over the set of queries issued as characteristic queries. The accuracy for content-based or performance-based methods was calculated over all the queries. Because the queries of "broad" categories and those of "narrow" categories yielded similar conclusions, the figures for "narrow" categories are omitted.

As shown in Figure 5.2, initial hub selection without user modeling (content/performance-based) underperformed that with user modeling (interest-based) due to the inability to identify uncharacteristic queries not related to search history. Because non-clustered interest-based initial hub selection didn't distinguish between different interests although each hub's performance for different interests was likely to be different, it was less effective than clustered interest-based initial hub selection. The superior performance of interest-based initial hub selection with query clustering resulted from its ability to measure each hub's performance for different interests separately, and to use the measured performance for hub selection when (and only when) past search could effectively guide future search.

6. APPROACH ROBUSTNESS

The previous section showed that our approach can provide significant gains in federated search performance with a certain set of parameter values. In this section we study the robustness of our approach by exploring the impact of different parameter settings. When not mentioned, the default parameter values are those shown in Table 5.1 with the exception that $ttl_{characteristic}$ is set to 0 and m is set to 5 (to focus on the performance of initial hub selection without further hub routing). Since similar conclusions can be drawn using either set of queries, only the results for the queries of "broad" categories are shown due to space constraints. A vertical dashed line in each figure of this section marks the default value of the corresponding parameter.

First we investigate whether the performance of our approach is sensitive to the values of the classification threshold $T_{classify}$ and the clustering threshold $T_{cluster}$. In theory, a tighter classification threshold causes more queries to be issued as uncharacteristic queries with a large search radius, which results in lower search efficiency but can reach a higher percentage of the hubs. Correspondingly, a looser classification threshold increases search efficiency with the possibility of hurting search accuracy. As to the clustering threshold, clusters created using a tighter clustering threshold represent narrower topics so that the percentage of the queries issued as uncharacteristic queries is likely to increase, resulting in lower efficiency but potentially higher accuracy. A looser clustering threshold leads to less cohesive query clusters representing broader topics with higher search efficiency. Therefore, the classification threshold and the clustering threshold have similar effects on search accuracy and efficiency. Figure 6.1 shows the change in search performance as the value of $T_{classify}$ varies from 6.5 to 7.5 (with a default value of 7.0). The left vertical axis denotes the percentage of the queries, which is a rough measure of search efficiency since efficiency is linearly correlated with the percentage of the queries issued as characteristic queries. Search accuracy is measured by the mean average overlap precision over the set of queries issued as characteristics queries (the right vertical axis). As expected, loosening the default threshold value increases search efficiency (the solid curve) moderately due to a larger percentage of the queries being issued as characteristic queries with a small search radius. However, the improvement in efficiency is not at the cost of significantly deteriorating search accuracy (the dotted curve). Overall, the figure indicates that the performance of our approach is quite robust when the threshold value is chosen within a certain range. Similar results can be obtained for the clustering threshold $T_{cluster}$ which varies from 1.0 to 2.0 (with a default value of 1.5). Its figure is omitted for space reasons.

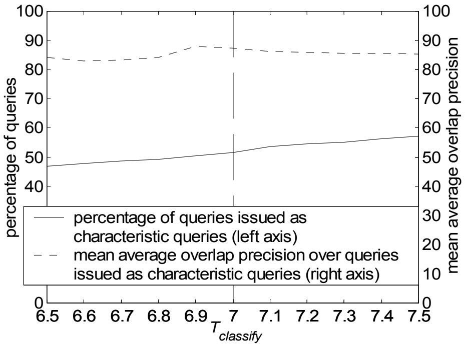

Figure 6.1 The change in performance as $T_{classify}$ varies.

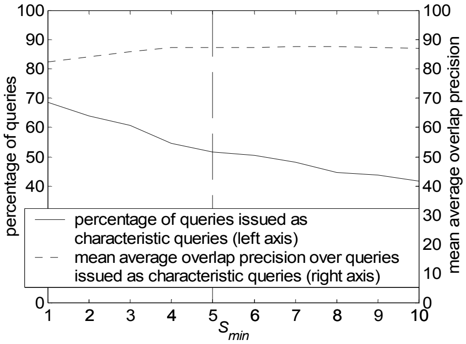

Figure 6.2 The change in performance as S_{min} varies.

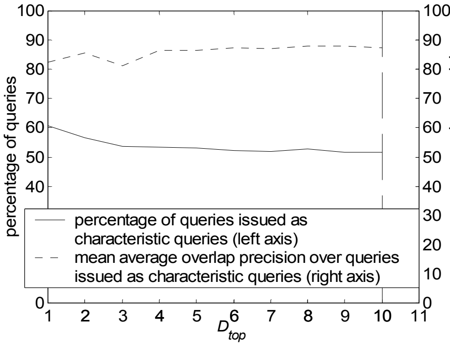

Figure 6.3 The change in performance as D_{top} varies.

As discussed in Section 3.2, our approach requires each query cluster to reach a certain size S_{min} before it is regarded as a topic of interest. Figure 6.2 depicts the results when S_{min} varies from 1 to 10 (with a default value of 5). From the figure we can see that the gain in search accuracy levels off after S_{min} reaches 5 but search efficiency continues to decline as S_{min} increases. Therefore, 5 queries are sufficient to represent a persistent interest, indicating that our approach doesn't require a large amount of training data to learn the different interests of the user.

Another parameter to consider is the number of the top-ranked merged documents D_{top} used to generate a query's representation for clustering and to measure a hub's performance in locating relevant contents. The results obtained by varying D_{top} from 1 to 10 (with a default value of 10) are shown in Figure 6.3. The figure shows that a too-small D_{top} value results in unreliable search performance. This is because although relevant documents are very likely to be ranked more highly than most non-relevant documents, they may fail to appear among the very few top-ranked merged documents. To achieve reliable performance, the set of the top-ranked merged documents needs to be sufficiently large to guarantee the inclusion of relevant documents. As shown in the figure, both search accuracy and search efficiency become quite stable after the value of D_{top} reaches 5. This indicates that when feedback from the user is not available and "pseudo-relevance feedback" has to be used, at least 5 top-ranked merged documents are required for the performance to be robust.

The maximum number of recorded query clusters N_{max} is also a factor that can affect the performance of our approach. The experimental results of using different N_{max} values (omitted here due to space constraints) show that within the tested range of 10 to 200, when N_{max} is smaller than 40, existing query clusters need to be constantly removed in order to make room for new clusters. The high turnover rate prohibits useful clusters that represent the user's persistent interests from being formed and becoming stabilized. As a result, the percentage of the queries issued as characteristic queries is small and the search accuracy of these queries is low due to the low quality of clustering. Therefore, to avoid negatively affecting the effectiveness of query clustering, the constraint on the maximum number of recorded query clusters cannot be too tight.

7. CONCLUSIONS

User modeling has mostly been studied for information filtering and Web retrieval to improve the system's performance in delivering relevant documents. In this paper, we explore its use for a new task under a new type of environment. Specifically, we develop an approach to modeling user interests for improving the efficiency of full-text federated search in peer-to-peer networks without degrading its accuracy. By using past queries and ranked search results to model the user's persistent interests and evaluate the performance of each directory service with regard to each interest, an information consumer is able to make effective interest-based initial directory service selection, which can significantly reduce the amount of query routing required. In our study, we find that our approach works effectively in an unsupervised manner without requiring a large amount of training data, and it is robust to a range of parameter values.

Compared with previous work on user modeling for information retrieval, our approach has several distinctive characteristics. First, it explicitly distinguishes between queries that are closely related to the user's persistent interests and queries that aren't, which allows different search strategies to be applied to different kinds of queries. Second, by automatically creating new clusters to model new interests as they arise and constantly reassessing old ones, our approach tracks evolving user interests in a timely manner, and naturally captures not only long-term interests that span months and years, but also short-term interests that last for only several days. Third, in contrast to most of the previous research on user modeling that focuses on detailed user models for tasks such as filtering and collaborative search, our work shows that coarse user models, which can be learned from very small amounts of training data, can be useful for some retrieval tasks. This reminds us that different tasks may require user models of different granularity, and the most appropriate approach to user modeling for a particular task is not necessarily the one that generates the finest-grained user models.

ACKNOWLEDGMENTS

This material is based on work supported by NSF grant IIS-0240334. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsor.

REFERENCES

[1] R. Baeza-Yates. Web mining in search engines. In *Proc. of the 2004 Conference on Australasian Computer Science*.

[2] A. Crespo and H. García-Molina. Semantic overlay networks for P2P systems. Technical report, Computer Science Department, Stanford University, 2002.

[3] F. Diaz and J. Allan. Browsing-based user language models for information retrieval. Technical Report, CIIR, University of Massachusetts, Amherst, 2003.

[4] S. Gauch, J. Chaffee and A. Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent System*, 1(3), 2003.

[5] N. Glance. Community search assistant. In *Proc. of the 2001 International Conference on Intelligent User Interfaces*.

[6] R. Krovetz. Viewing morphology as an inference process. In *Proc. of SIGIR 1993*.

[7] A. Löser, F. Naumann, W. Siberski, W. Nejdl and U. Thaden. Semantic overlay clusters within super-peer networks. In *Proc. of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 2003*.

[8] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proc. of CIKM 2003*.

[9] J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *Proc. of ECIR 2005*.

[10] J. Lu. Full-text federated search in peer-to-peer networks. Technical report CMU-LTI-05-197, Language Technologies Institute, Carnegie Mellon University, 2005.

[11] M. Ramanathan, V. Kalogeraki and J. Pruyne. Finding good peers in peer-to-peer networks. In *Proc. of IPDPS 2002*.

[12] S. Ratnasamy, S. Shenker and I. Stoica. Routing algorithms for DHTs: Some open questions. In *Proc. of the 2002 International P2P Workshop*.

[13] S. Robertson and D. Hull. The TREC-9 Filtering track report. In *Proc. of TREC 2001*.

[14] Y. Shao and R. Wang. BuddyNet: history-based P2P search. In *Proc. of ECIR 2005*.

[15] X. Shen, B. Tan and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proc. of SIGIR 2005*.

[16] K. Sripanidkulchai, B. Maggs and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proc. of Infocom 2003*.

[17] M. Speretta. Personalizing search based on user search histories. In *Proc. of CIKM 2004*.

[18] K. Sugiyama, K. Hatano and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. of WWW 2004*.

[19] J. Teevan, S. Dumais and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of SIGIR 2005*.

[20] E. Voorhees, N. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proc. of SIGIR 1995*.

[21] J. Wen, J. Nie and H. Zhang. Query clustering using user logs. *ACM Transactions on Information Systems*, 20(1), 2002.