

Network Flow for Collaborative Ranking

Ziming Zhuang¹, Silviu Cucerzan², and C. Lee Giles¹

¹ Information Sciences and Technology, The Pennsylvania State University,
University Park, PA 16802, U.S.A

² Microsoft Research, One Microsoft Way
Redmond, WA 98052, U.S.A

{zzhuang, giles}@ist.psu.edu, silviu@microsoft.com

Abstract. In query based Web search, a significant percentage of user queries are underspecified, most likely by naive users. Collaborative ranking helps the naive user by exploiting the collective expertise. We present a novel algorithmic model inspired by the network flow theory, which constructs a search network based on search engine logs to describe the relationship between the relevant entities in search: queries, documents, and users. This formal model permits the theoretical investigation of the nature of collaborative ranking in more concrete terms, and the learning of the dependence relations among the different entities. *FlowRank*, an algorithm derived from this model through an analysis of empirical usage patterns, is implemented and evaluated. We empirically show its potential in experiments involving real-world user relevance ratings and a random sample of 1,334 documents and 100 queries from a popular document search engine. Definite improvements over two baseline ranking algorithms for approximately 47% of the queries are reported.

Keywords: Graph models, Network flow, Graph theory, Collaborative ranking, User feedback, FlowRank.

1 Introduction

The intuition behind collaborative ranking is that in the context of Web search collective knowledge enhances the ranking of search results. A number of methods to facilitate collaborative ranking have been proposed and some effectively implemented, particularly in online recommendation systems. By introducing a formal model, this work systematically investigates the nature of collaborative ranking.

Why does collaborative ranking work? Given the context of relevant queries, naïve queries can be improved by looking at relevant and articulated queries (and corresponding search results) from other users. By observing the collective patterns of access, the ranking system learns to favor certain search results. For the most part, existing collaborative ranking algorithms take into account the intricate relationships between relevant queries, documents, and collaborators. We propose a graph algorithmic model to study such relationships. Based on this model, a novel *FlowRank* algorithm translates the collaborative ranking problem into a network flow problem. The heterogeneous interactions between queries, documents, and collaborators are drawn into a concise and cohesive framework. Our paper makes the following contributions:

- We introduce a formal model of collaborative ranking, which mathematically describes the interactions between users, queries, and documents, and relates the ranking problem with the network flow problem for which there is a large body of work.
- We propose a practical algorithm for collaborative ranking based upon our model, and empirically demonstrate its potential in a preliminary experiment using real-world data.

The rest of the paper is organized as follows¹. In Section 2, we briefly review a number of relevant studies in collaborative ranking and graph modeling of the Web. We describe in detail the algorithmic model and propose a derived ranking algorithm, *FlowRank*, in Section 3. In Section 4, we present an evaluation involving realistic search scenarios. We conclude our paper with plans for future work in Section 5.

2 Related Work

There has been a considerable amount of work on the topic of collaborative search and ranking. We will discuss succinctly the major body of work that directly relate to ours, in particular to graph models of the Web and collaborative ranking algorithms.

The Web graph [1, 2] has been explored with numerous studies focusing on the hyper-linked structure in order to aid web search [3, 4, 5, 6, 7]. Flow-based algorithms have been proposed to identify and mining online communities [8, 9], to perform clustering [32], and to identify the bottlenecks in a Markov decision process [10]. In particular, Chitrapura and Kashyap [11] proposed a flow-based model for document ranking, which uses the network flows in a search graph as a measurement of relevance. In their model, the volume of the flows indicating the degree of relevance of the nodes (documents) to the associated labels (queries) is used to compute a query dependent or independent ranking of the documents, which is similar to the underlying idea we propose. However, their model is a single-user model, while our model employs multiple collaborating users with various degrees of similarity to the target user and different relevance feedbacks.

Collaborative search and ranking is in a way similar to meta-search; leveraging naïve and advanced users, the original search results are re-ranked based on collective knowledge. The search system can take into account the expertise level of the users through user profiling [12], biasing the relevance ranking by using such profiles. When a user is not satisfied with the search results, similar queries submitted by other users can be used to expand or improve the original query [13]. User clickthrough has been shown as an accurate reflection of users' preferences on the retrieved pages [14], to suggest search intentions [15] and similar queries and documents [16]. A hit matrix, which records users' clickthrough to pages, is implemented in the I-SPY system [17]. By using collective clickthrough as an indicator of the likeliness of web pages to be visited by users, the system estimates the pages' relevance to a given query, and re-ranks the pages based on the learned preferences. One of the most recent evaluations of collaborative search involves studying the behavior of actual users [18].

¹ Due to space constraints, an extended version of this paper discussing the generalization of the proposed model is available for download on the first author's website.

Our work builds upon the idea of graph modeling of the Web search problem, in a framework of relevance ranking of documents based on the associated flows of information. However, to the best of our knowledge, there is no existing work that is similar to our approach of constructing the graph model and the derived collaborative ranking algorithm.

3 The *FlowRank* Model

3.1 Motivation

Consider the following scenario of a typical collaborative search: A user u_a searches the Web for query q_o '*Olympic national park*', and a document set D_o has been retrieved. Then, suppose there is a *collaborator* u_b who searches for a relevant query q' '*camping hiking Olympic*', and a document set D' has been retrieved. We define collaborators to be those users whose queries and search results may be of interests to u_a with respect to u_a 's current search goal, regardless of whether they perform the search in a synchronous or asynchronous fashion, or whether they are aware of each other. How relevant u_b 's query is to u_a , and how similar u_b is to u_a in terms of interests and preferences, should be considered in the case of collaborative search. If u_b frequently accesses (which may mean, for example, that the user clicks on) some of the returned documents, it could indicate that these documents are of high quality and are preferable to other documents. Thus u_a might also prefer these on the condition that u_b 's query is somewhat similar and u_b has similar interests or preferences.

From the above scenario, we see that a complete model for collaborative ranking needs to consider all of the following: the similarity between the original query (q_o) and the retrieved documents (D_o), the similarity between the original query (q_o) and other relevant queries (q'), the similarity between the relevant queries (q') and the corresponding documents (D'), the similarity between the user (u_a) and the collaborators (u_b), and the access patterns of the users. All of these factors combined could contribute to the relevance judgment of the documents retrieved for a given query, and can be employed in a collaborative scheme of ranking those documents.

We have devised a formal model of collaborative ranking, in which all these factors are accounted for in a unified formal framework that permits the study of correlated search events. We accomplish this by investigating the network flows of a transformed query graph. In our model, each retrieved document is represented by a sink node in the graph and associated with a flow value. The flow values are bounded by the various capacities of the arcs, which correspond to the factors discussed in the previous paragraph (details to follow in the next section). When the flow network becomes saturated, the values of the obtained flows associated with the documents are used to re-compute the collaborative relevance ranking.

3.2 Graph Definition and Transformation

A query graph visualizes relevant entities (queries, documents, and users) in a given collaborative search setting. We transform this graph into a network in which arc capacities encode the relationships between the entities. In this section we formally describe the steps to derive the graph model. An algorithm for collaborative ranking based on this model will be introduced in the next section.

We first discuss the graph structure that we associate with a user query, which links users, queries, and documents sets, denoted as U , Q , and D respectively.

Assume that a target user u^i , submits a target query q^i , for which a set of documents $D^i = \{d_j^i \mid i=1..m\}$ matching the target query exists. Let $Q^i = \{q_j \mid \exists j \in [1..n], \text{arc}(q_j, d_j^i) \neq \text{null}\}$ be the set of all user queries that retrieve at least one of the documents retrieved by the target query q^i . Let $U^i = \{u_j \mid \exists q \in Q^i, \text{arc}(u_j, q) \neq \text{null}\}$ denote the set of users who submitted the query that retrieved a document in D^i . Let G denote the whole search graph with no constraint. We denote by G_i the subgraph of G with the vertices $D^i \cup Q^i \cup U^i$ and all arcs incident in these vertices from G . Intuitively, the graph G_i includes those users who sent to the search engine the same query or a different query that retrieves at least one of the documents retrieved by the target query. Users who have not submitted q^i and queries that do not retrieve any document in D^i will not be included in this graph. The goal is to rank collaboratively the documents in D^i by considering the *appropriately relevant* users, queries, and documents. Note that in practice, there can be various definitions for $q_j \in Q^i$. For example, q_j can be an expansion of the target query q^i [19] or a query submitted by a user that searched for q^i in the same search session and/or within a certain interval of time. Figure 1 (a) depicts a sample query graph G_i .

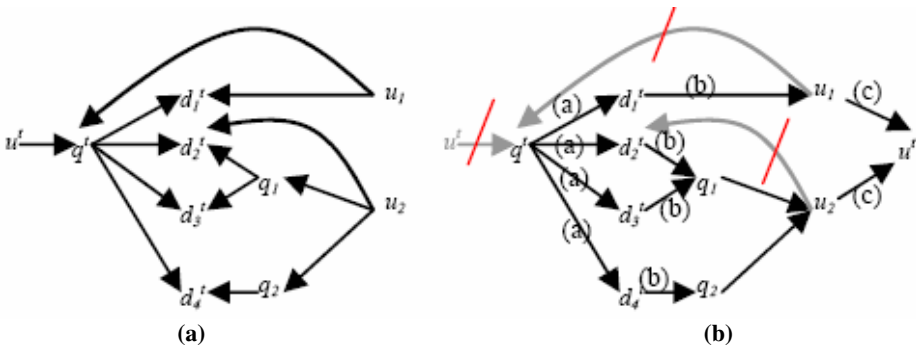


Fig. 1. (a) A sample query graph G_i , (b) An example of G' constructed from G_i ,

We then construct G' based on G_i . Figure 1 (b) depicts G' constructed from G_i . There are three types of capacities assigned to the arcs in G' : type (a) expresses the relevance between the target query and the documents; type (b) reflects various degrees of collective endorsement to the documents by the collaborators, either through the original query or alternative queries; type (c) specifies the relationship between the targeted user and his collaborators. **In particular:**

- Some arcs have their directions reversed in order to form the cascade of bounds for the network flows, i.e. type (a) capacities are more important than type (b); type (b) are more important than type (c). This is arguably reasonable because in ranking the relation between the target query and the documents should always be considered first, while the collective endorsement by the collaborators is counted as an assistive measure. Please see the following algorithm for details on which arcs to be reversed.

- Steps (2)~(6) of the following algorithm provide a specific computation recipe of the arc capacities. However, alternative strategies/heuristics can be employed to weigh the relationships among the different entities.

Let $U' = U - \{u'\}$ denote all users in this graph except the target user and $Q' = Q - \{q'\}$ the set of all user queries in G , excluding the target query q' . We construct G' as follows:

- (1) Remove the arc from u' to q' . The relationship between u' and q' will be implicitly expressed in the generated flow values (defined in Section 3.3).
- (2) Add an arc between any $u_i \in U'$ and u' , and define its capacity $C(u_i, u')$ to be a similarity score between the two users, $0 < C(u_i, u') \leq 1$. This places an upper-bound for the flow values proportional to the user-user similarity.
- (3) Define the capacity between q' and any $d_k \in D'$ to be a value in $[0, 1]$ that indicates a matching score between the target query q' and the document d_k . This places an upper-bound for the flow values proportional to the query-document similarity.
- (4) For any $u_i \in U'$ and $q_j \in Q'$, if $arc(u_i, q_j) \neq null$ (meaning that the user u_i submitted query q_j), reverse the direction of this arc and set the capacity $C(q_j, u_i) = 1$, so that the flow value on $arc(q_j, u_i)$ is bounded *only* by the *upstream* capacities (i.e. query-document similarity), because of the reversed arc direction. Otherwise, set the capacity $C(q_j, u_i) = 0$. At this point we do not consider the frequency of u_i submitting query q_j .
- (5) For any arc $arc(q_j, d_k)$ from q_j in Q' to a document $d_k \in D'$, reverse the direction of this arc and define the capacity $C(d_k, q_j)$ to be $C(d_k, q_j) = sim(q_j, d_k) \cdot sim(q_j, q') \cdot P(d_k | U_{j,k})$, where $sim(q_j, d_k)$ in $[0, 1]$ indicates a matching score between the query q_j and the document d_k ; $sim(q_j, q')$ indicates how similar this alternative query q_j is to the target query q' ; $P(d_k | U_{j,k})$ indicates the conditional probability of users visiting d_k given that they submitted the query q_j , which can also be written as:

$$C(d_k, q_j) = sim(q_j, d_k) \cdot sim(q_j, q') \cdot \frac{Click(U_{j,k}, d_k)}{Click(U_{j,k}, D')}$$

$U_{j,k} = \{u_i \in U' \mid arc(u_i, q_j) \neq null \wedge arc(q_j, d_k) \neq null\}$. Note that $U_{j,k}$ typically contains more than one user; $Click(U_{j,k}, d_k)$ is the total number of clicks made by the users who submitted q_j on d_k ; $Click(U_{j,k}, D')$ is the total number of clicks of these users on the whole search result set. $C(d_k, q_j)$ places an upper-bound for the flow value jointly decided by the similarity between an alternative query q_j and the target query q' , the matching score between q_j and the document d_k , and the likelihood that a user visits d_k given that he submits q_j .

- (6) For any $arc(u_i, q')$ for which $\exists d_k \in D', arc(u_i, d_k) \neq null$, reverse the direction of $arc(u_i, d_k)$ and assign $C(d_k, u_i) = sim(q', d_k) \cdot P(d_k | u_i) \cdot [1 - P(d_k | \overline{\{u_i\}})]$, where the last two factors represent the probability of a user u_i visiting d_k and the conditional probability of the *other* users *not* visiting d_k given that they submitted the target query q' . This can also be written as:

$$C(d_k, u_i) = \text{sim}(q^t, d_k) \cdot \frac{\text{Click}(u_i, d_k)}{\text{Click}(u_i, D^t)} \cdot \left(1 - \frac{\text{Click}(\overline{\{u_i\}}, d_k)}{\text{Click}(\overline{\{u_i\}}, D^t)} \right)$$

where $\overline{\{u_i\}} = U - \{u_i\}$; $\text{Click}(u_i, d_k) = 1$ if u_i clicked on d_k and 0 otherwise; $\text{Click}(u_i, D^t)$ is the number of documents clicked by u_i . $C(d_k, u_i)$ places an upper-bound for the flow value jointly decided by the matching score between the target query q^t and the document d_k , and the likelihood that other users also visit d_k given that they also submit q^t .

(7) Remove all the arcs from $u_i \in U$ to q^t . Because the inherent relationship is already reflected in the definition of $C(d_k, u_i)$, these arcs are redundant and do not contribute to the collaborative ranking process.

(8) END

Finally, we want to provide an intuitive justification for constructing G' . A query represents the information needs of a user, and the motivation of search is to satisfy such needs. In G' , flows of relevant information are moving through a network of collaboration towards the target user via different routes. Each of these flows contributes as a part to the overall information gain pertinent to the user's needs. On each of the routes, there are an appropriately relevant document, an appropriately relevant query, and an appropriately relevant user; all of them together determine the contribution of the associated flow. Flows, when saturated, represent the collaborative contribution to the overall satisfaction of the target user's information needs. By investigating G' , we are able to study the relationship between the collaborative entities when the flows are sorted in the order of their contributions to the overall satisfaction of the target user's information needs. Accordingly the associated documents en route can also be ranked using the *FlowRank* algorithm described in the next section.

3.3 The *FlowRank* Algorithm

Let s denote the source and t denote the sink, $F_{\max}(s, t)$ is defined as the maximum flow that can be routed from s to t , which obeys all the capacity constraints. Intuitively, if the arcs are water pipes, the vertices are where they join each other, and the capacities on the arcs represent the cross-sectional area of the pipes, to find the maximum flow is to find how much water can be moved from s to t , given the constraints of the cross-sectional area of the pipes. The *FlowRank* algorithm is described in Figure 2.

The *FlowRank* algorithm is based on the well-known Maximum Flow - Minimal Cut theorem [20]. This theorem proves that the maximum flow of a given network is equal to the minimal cut that separates the source and the sink, which in our case the cut is D' , proved as follows. A set of cut-vertex denotes the set of vertexes whose removal will disconnect the graph [30]. By assumption that D' is not the set of cut-vertex on G' , we remove D' and its related arcs, and still have q^t connected to G' via a set of vertex V_{q^t} . $\forall v_{q^t} \in V_{q^t}: v_{q^t} \notin Q$, because by definition there's no arc exists between two queries; $v_{q^t} \notin U$, because of step (4) and (5) in 3.2; thus $v_{q^t} \notin D$. Because $D = D' \cup D''$ and D' is removed in step (1) in 3.2, $v_{q^t} \in D''$. But this contradicts the assumption that D' has already been removed from G' . Q.E.D.

The above proof is important for the validity of the algorithm in that $\forall F(q^t, u^t), \exists d_i^t \in D^t$ which is on the path of $F(q^t, u^t)$. Because D^t is not inter-connected, we conclude that there is d_i^t on each of the paths of $F(q^t, u^t)$, so that D^t can be ranked by sorting $F(q^t, d_i^t)$.

Algorithm. FlowRank

Input: Graph G^t
Output: Permutation $\pi \{ \pi(d_1^t), \dots, \pi(d_m^t) \}$, $d_1^t \dots d_m^t \in D^t$.
 1: $F_{max}(q^t, u^t) \leftarrow$ the maximum flow from source q^t to sink u^t
 2: **foreach** document $d_i^t \in D^t$
 3: $F(q^t, d_i^t) \leftarrow$ the flow value from source q^t to sink d_i^t
 4: Sort (descending) the documents d_i^t in D^t using $F(q^t, d_i^t)$
 5: **return** ranks $\{ \pi(d_1^t), \dots, \pi(d_m^t) \}$

Fig. 2. The *FlowRank* Algorithm

4 Evaluation

4.1 Experiment Setup

From the query logs of a popular scientific document search engine, The *CiteSeer* Digital Library (<http://citeseer.ist.psu.edu>), we extracted a random sample of 100 queries and the associated 1,334 unique documents retrieved by the search engine as relevant results. Obtaining editorial ratings for a large number of documents is an extremely time-consuming and labor-intensive process, however the size of our dataset is comparable to those used in [21]. The queries were anonymized, and then verified by human annotators to be *meaningful* [22].

Our evaluation required a log of user interactions (i.e. clickthroughs, etc.) with the documents. The extracted queries and documents were presented to five evaluators, all graduate students in Computer Science. For each document, the title and abstract/snippet were displayed. In order to minimize the potential bias induced by ranking, for each query the associated documents were shuffled so that the evaluators were not aware of the original ranking produced by the search engine, and the evaluators were explicitly informed about this process. The evaluators were asked to independently choose one of the three ratings for each document based on their subjective perception of how relevant a document is to the associated query. The three ratings were: “*Definitely clicked*”, which means the evaluator believes he/she would click on the document if he/she were to submit the query; “*Probably clicked*”, which means the evaluator may or may not click on the document, and “*Never clicked*”, which means the evaluator believes he/she would not click on the document. The data collected in this procedure is labeled as D_c . Ideally we would like to keep track of a user’s clicks on a document, denoted by a 4-tuple $(q_id, doc_id, u_id, p_click)$, where q_id and doc_id denote the query and corresponding documents, u_id denotes the user, and p_click denotes the probability of the user clicking on the document. In

processing D_s , the rating “Definitely clicked” was translated into $p_click=1.0$; “Probably clicked” was translated into $p_click=R$ where R is a random variable within $(0, 1)$; “Never clicked” was translated into $p_click=0.0$. The obtained data D_s was then used to simulate user clickthrough.

Relevance judgments for the dataset were generated by presenting the same queries and documents to another two evaluators who are both computer scientists. This time, the evaluators could check the actual content of the document when necessary. Each of the evaluators was asked to rate the document in a five-point scale from 0 to 4, defined as:

0. *Irrelevant match*: the document did not contain any information about the query.
1. *Marginally relevant match*: the query terms might appear in the document but it was mainly about something else.
2. *Borderline match*: the document could be rated as 1 or 3.
3. *Fairly relevant match*: the document contained relevant information about the query terms, but should not be picked if only one document were allowed.
4. *Best match*: the document contained highly relevant information about the query terms, and could be picked if only one document were allowed.

Data collected in this phase is labeled as D_e .

A number of measures have been proposed to quantitatively describe the similarity between queries [15, 24, 25, 27]. In our implementation, the definition of similarity between queries was similar to the *common query title* measurement as described in [25]:

$$sim(q_i, q_j) = \frac{|T_i \cap T_j|}{\max(|T_i|, |T_j|)}$$

where q_i, q_j were two queries and T_i, T_j were the terms in the titles of the documents returned by the search engine. In other words, the similarity between the two queries was in proportion to the number of common terms in the titles of the search results. If the titles were similar, the queries were also similar.

User similarity metrics are commonly adopted in collaborative recommendation and filtering systems, because by learning from other “like-minded” users, one can predict a user’s preference (see [25] for a recent survey). User similarity is inherent in our model because of step (2) of the algorithm. The user similarity score is the upper-bound for the flow values. For simplicity of computation, we currently treat all users the same, so for any two users u_i and u_j , $sim(u_i, u_j) = 1$. We justify this assumption by noting that all of the five evaluators were graduate students in Computer Science and most likely familiar with the search topics. However, it would be interesting to observe the impact of different user similarity metrics in the next phase of evaluation.

Finally, the similarity scores between the queries and documents were derived from the search engine. Now a transformed query graph was computed for each of the queries, and the *FlowRank* algorithm was implemented on the graphs to generate the collaborative rankings. Adjacent matrixes were built to represent the transformed query graphs, with elements being the arcs’ capacities. In our experiment, the maximum-flow calculation module was based on Rothberg’s implementation of Goldberg’s Push-Relabel algorithm [23], which is usually considered the fastest in practice.

4.2 Results and Discussion

The output of *FlowRank* (F) was compared with that of other two baseline ranking algorithms for evaluation. The HITS algorithm [4] was implemented and the ranking (H) for the documents was recorded. The original ranking (S) generated by *CiteSeer* was also recorded. The metrics to measure the ranking accuracy was the Discounted Cumulative Gain (DCG) first introduced in [26] and compared to other metrics in [28]. The main reason to use DCG is that it assigns more weights to highly ranked documents, and allows us to differentiate various levels of subjective relevance judgment for the human evaluators. For a given query q , DCG is defined as:

$$DCG(q, N) = \sum_{d=1}^N \frac{(2^{R(d)} - 1)}{\ln(1 + d)} \quad (1)$$

where $R(d)$ is the editorial rating of the d 'th webpage in the top N search results.

Intuitively, a higher DCG reflect a better ranking of the search results. DCG was computed for the top 20 ranked documents in H , S , and F , since users seldom look beyond the first few result pages [31]. The ratings from the two evaluators in D_c were averaged as the editorial ratings to be used in the DCG calculation.

FlowRank achieved significant improvements on the rankings produced by the two baseline algorithms. This advantage was reflected in the average DCG metrics. In both comparisons, *FlowRank* was able to improve the ranking for about 47% of the queries. Over all ranks, the average DCG for F was 59.42, compared with 57.54 for S and 57.37 for H . The number of queries with DCG increased, decreased, and without change was summarized in Figure 3(a).

Figure 3(b) plots the average DCG curves for the three ranking algorithms (F , H , S) at ranks 1 – 20, together with the ideal curve. *FlowRank* outperformed the other two baseline algorithms and quickly approached the ideal curve, which began to level off upon the rank 10. This confirmed that the documents ranked below position 10 were in fact less relevant.

In order to further investigate if there is linear correlation between the collaborative contribution considered by *FlowRank*, and the amount of increase in the average DCG over the two baseline algorithms, the improvement in the DCG scores for each of the queries was cross-examined with the size of the transformed graph. Here an assumption was made that the larger the transformed query graph, the more collaborative the search process would be. A correlation test was performed on the two variables: size S and increase in the DCG I . The Pearson correlation of S and I was 0.334 with a P-value of 0.033 (α -level was 0.05), confirming a positive correlation between the two.

A simplification was drawn in the current implementation of the flow-based model, in which the arc capacities between $\{u_i\}$ and u_i were assigned the same values, indicating identical collaborators. Three runs of the *FlowRank* algorithm were performed in which the number of users was purposefully changed. Correlated changes in the DCG improvement had been observed. The correlation was not significant, which we believed would be mostly due to the small number of users.

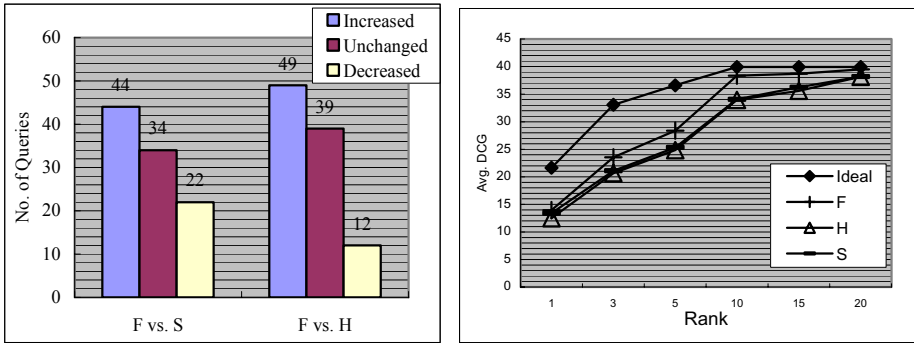


Fig. 3. (a) DCG changes for queries; *FlowRank* was able to improve the search ranking for about 47% of the queries. (b) Average DCG curves; The *H* and *S* curves hang below the *F* curve by about 15%.

The maximum flows were currently calculated using the Push-Relabel algorithm [23], because the sizes of the graphs were relatively small, and runtime efficiency was a primary concern due to the online nature of collaborative ranking. However, depending on the scale (and often the subject domain) of a collaborative search, the graph can become very dense or sparse, or can become so huge that access to the entire graph is impractical. Different maximum-flow algorithms, e.g. the shortest augmentation path algorithm [29], should be used with case-by-case consideration.

5 Conclusion and Future Work

We proposed a comprehensive flow-based graph model of collaborative ranking. Exploiting the relationships between relevant queries, documents, and users, and users' access patterns on the retrieved documents, this model translates the collaborative ranking problem into a flow-calculation problem in a search network. This unique perspective considers the ranking problem as a graph flow problem for which there is a large body of work. We discussed the implications of several ranking scenarios and presented a derived practical ranking algorithm. Evaluations in the document search domain using the DCG metrics showed its effectiveness by measurable improvements over the two baseline ranking algorithms.

There are a number of directions towards which this work can be applied. Future work can explore different similarity metrics and use the *FlowRank* model to quantitatively investigate the complicated relationships between queries, documents, and users in a collaborative search setting. Obtaining editorial ratings for a large number of documents is an extremely time-consuming and labor-intensive process, which limited the scale of our preliminary experiment. A reasonable next phase of evaluation is to compare *FlowRank* against other collaborative ranking algorithms using a large-scale dataset. Instead of using simulated clickthrough data, real-world clickthrough logs can be used so that the definitions of the arc capacities can be further fine-tuned. Additionally, it may be valuable to also take into account user

interactions other than clickthroughs, such as how long a user spent reading a document. Given the generalizable nature of the proposed network flow model, we believe that it can be applied to a number of other problems dealing with user feedback and collaborative search behavior.

Acknowledgments. We gratefully acknowledge E. Brill, C. Burges, M. Richardson, and the anonymous reviewers for their comments, Y. Sun for his help on the experiments, and partial support from the National Science Foundation.

References

1. Bharat, K., Broder, A., Henzinger, M., Kumar, P., and Venkatasubramanian, S. The Connectivity Server: Fast Access to Linkage Information on the Web. In Proc. of the 7th International World Wide Web Conference, pp. 469-477, 1998.
2. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. Graph structure in the web. In Proc. of the 9th International World Wide Web Conference, pp. 309-320, 2000.
3. Carriere, J., and Kazman, R. WebQuery: Searching and visualizing the Web through connectivity. In Proc. of the 6th International World Wide Web Conference, 1997.
4. Kleinberg, J. Authoritative sources in a hyperlinked environment. *J. of the ACM*, Vol. 46, Issue 5, pp. 604-632, 1999.
5. Brin, S., and Page, L. The anatomy of a large scale hypertextual web search engine. In Proc. of the 7th International World Wide Web Conference, pp. 107-117, 1998.
6. Chakrabati, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., and Rajagopalan, S. Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text. In Proc. of the 7th International World Wide Web Conference, 1998.
7. Tomita, J., and Kikui, G. Interactive Web Search by Graphical Query Refinement. In Proc. the 10th International World Wide Web Conference, 2002.
8. Flake, G. W., Lawrence, S., and Giles, C. L. Efficient identification of web communities. In Proc. of the 6th International Conference on Knowledge Discovery and Data Mining, pp. 150-160, 2000.
9. Flake, G. W., Tsioutsoulis, K., and Zhukov, L. Methods for Mining Web Communities: Bibliometric, Spectral, and Flow. *Web Dynamics*, Springer Verlag, 2003.
10. Menache, I., Mannor, S., and Shimkin, N. Q-Cut - Dynamic Discovery of Sub-goals in Reinforcement Learning. In Proc. of the 13th European Conference on Machine Learning (ECML), pp. 295-306, 2002
11. Chitrapura, K. P., and Kashyap S. R. Node Ranking in Labeled Directed Graphs. In Proc. of ACM Conference on Information and Knowledge Management, pp. 597 - 606. 2004.
12. Chidlovskii, B., Glance, N. S., and Grasso, M. A. Collaborative Re-Ranking of Search Results. In The National Conference on Artificial Intelligence 2000 Workshop on AI for Web Search, pp. 18 - 23. 2000.
13. Zaiane, O. R., and Strilets, A. Finding Similar Queries to Satisfy Searches based on Query Traces. In Proc. of the International Workshop on Efficient Web-Based Information Systems, pp 207-216, 2002.
14. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G. Accurately Interpreting Clickthrough Data as Implicit Feedback. In Proc. of Annual ACM Conference on Research and Development in Information Retrieval (SIGIR) 2005, pp. 154-161, 2005.

15. Daume, H., and Brill, E. Web search intent induction via automatic query reformulation. In *Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics*, 2004.
16. Wen, J., Nie, J. and Zhang, H. Clustering user queries of a search engine. In *Proc. of the 10th International World Wide Web Conference*, pp. 162-168. 2001.
17. Freyne, J., Smyth, B., Coyle, M., Balfe, E., Briggs, P. Further Experiments on Collaborative Ranking in Community-Based Web Search. *Artificial Intelligence Review*, Vol. 21:3-4, pp. 229-252. 2004.
18. Smyth, B., Balfe, E., Boydell, O., Bradley, K., Briggs, P., Coyle, M., Freyne, J. A Live-User Evaluation of Collaborative Web Search. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*. 2005.
19. Vakkari, P. Subject knowledge, source of terms and term selection in query expansion. In *Proc. of the 24th European Conference in Information Retrieval*, 2002.
20. Ford, L. R. Jr., and Fulkerson, D. R. Maximal flow through a network. *Canadian J. of Mathematics*, Vol. 8, pp. 399-404, 1956.
21. Savoy, J. and Vrajitoru, D. Evaluation of learning schemes used in information retrieval. Technical Report CR-I-95-02, Faculty of Sciences, University of Neuchatel, 1996.
22. Risvik, K. M., Mikolajewski, T., and Boros, P. Query Segmentation for Web Search. In *Proc. of the 11th International World Wide Web Conference*. May 20-24, 2003.
23. Goldberg, A. V., and Tarjan, R. E. A new approach to the maximum-flow problem. *J. of the ACM*, Vol. 35, Issue 4, pp. 921-940. ACM Press, 1998.
24. Beeferman, D., and Berger, A. Agglomerative clustering of a search engine query log. In *Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 407-416, 2000.
25. Cheung, K., and Tian, L. Learning User Similarity and Rating Style for Collaborative Recommendation. *Information Retrieval*, Vol. 7: 3-4, pp. 395-410. Springer-Verlag, 2004.
26. Jarvelin, K., and Kekalainen, J. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proc. of the 23rd Annual ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 41-48. 2000.
27. Chien, S., Immorlica, N. Semantic Similarity Between Search Engine Queries Using Temporal Correlation. In *Proc. of the 14th International Conference on World Wide Web*, pp. 2-11. 2005.
28. Jarvelin, K., and Kekalainen, J. Cumulated Gain-based Evaluation of IR Techniques. In *ACM Transactions on Information Systems (TOIS)*, Vol. 20, No. 4, pp. 422-446, 2002.
29. Edmonds, J., Kapr, R. M. Theoretical improvements in the algorithmic efficiency for network flow problems. *J. of ACM*, Vol. 19, pp. 248-264. 1972.
30. Chartrand, G. Cut-Vertices and Bridges. *Introductory Graph Theory*. New York: Dover, pp. 45-49, 1985.
31. Jansen, J., and Spink, A. An Analysis of Web Documents Retrieved and Viewed, In *Proc. of the 4th International Conference on Internet Computing*, 2003.
32. Flake, G. W., Tarjan, R. E., and Tsioutsouluklis, K. Graph Clustering and Minimum Cut Trees. *J. of Internet Mathematics*, Vol. 1, No. 4, pp. 385-408. 2004.